MOTION CONTROL OF SERIAL-PARALLEL ROBOT MANIPULATORS

FINAL ROPORT

Shih-Liang Wang

September 4, 1996

U.S. ARMY RESEARCH OFFICE

DAAL 03-90-G-0185

North Carolina A&T State University

APPROVED FOR PUBLIC RELEASE;

DISTRIBUTION UNLIMITED.

SF 298 MASTER COPY KEEP THIS COPY FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	8-16-96	Final report, 8-16-90 - 8-14-96

4. TITLE AND SUBTITLE
Analysis of Spatial Mechanisms for Quasi-Direct Drive Redundant Robots

5. FUNDING NUMBERS
DAAL03-90-G-0185

6. AUTHOR(S)
Shih-Liang Wang

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)
Dept. of Mechanical Engineering
North Carolina A&T State University
Greensboro, NC 27411

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A serial-parallel robotic manipulator can be viewed as a closed mechanism with multiple-arms. It has the characteristics of both configurations: the high stiffness and accuracy of a parallel robot, and a large workspace and compact structure of a serial robot. Two serial-parallel robotic manipulators (a linkage robot and an articulated arm platform robot) were studied in this research on their direct and inverse kinematics, velocity, dynamics, and collision detection.

The direct and inverse kinematics problems were solved using the wrist position of each arm. The inverse velocity problem was solved by the force analysis and the principle of virtual work. Based on the velocity analysis, the singularity problem was researched, and the redundant actuation was analyzed. The dynamic models were established using the Lagrange formulation. An efficient algorithm was introduced to detect link collision. This algorithm can be extended to multiple cooperating robots as well. MATLAB was used to simulate robot motion and to verify various control algorithms.

To demonstrate the mobility and capability of a linkage robot, a prototype was built from off-the-shelf components whenever possible. Rhino robot's controller along with its motors is used to control this linkage robot.

14. SUBJECT TERMS	15. NUMBER IF PAGES
	54
	16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

C_{kj}^{i}	Coriolis and centrifugal terms at Joint i
$\mathbf{D}$	manipulator inertial tensor
D_{ij}	coupled inertia between Joints i and j
$\mathbf{D_k}$	kth link's inertial tensor
$\mathbf{F}$	external force
$\mathbf{I_k}$	inertial tensor of Link k
$\mathbf{J}$	position Jacobian matrix
L	line segment representing a link
$\mathbf{L_k}$	kth coordinate frame
$\mathbf{R}$	orientation matrix
$\mathbf{T}$	external torque
$T(q,\dot{q})$	kinetic energy
$\mathbf{T_{k-1}^{k}}$	the coordinate transformation matrix from frame k to frame k-1
a	link length
$\mathbf{a}$	approach vector
b_i	frictional torque of joint i
$\bar{c}^{j}$	coordinates of the center of mass of the jth link
c^{k}	coordinates of the center of mass of link k in the base coordinates
d	joint distance
l_i	distance from the tool to the wrist i
f_i	wrist force
g_0	gravitational constant
h_{ij}	distance from ith arm's jth joint to the wrist force f_{ij}
h_i	gravitational force at Joint i
k	distance between two wrists
m_k	mass of Link k
$\mathbf{n}$	normal direction vector
$\mathbf{o}$	orienting direction vector
$\mathbf{p}$	position vector
$\dot{q}$	joint rate
$\ddot{q}$	joint acceleration
s	parameter of a line segment
$\mathbf{t}$	joint torque
$\mathbf{u}$	unit vector in the direction of the wrist force
$\mathbf{v}$	linear velocity
$\mathbf{w}$	angular velocity

α	twist angle
θ	joint angle
τ_i	joint torque
ω_i	angular velocity

ABSTRACT

A serial-parallel robotic manipulator can be viewed as a closed mechanism with multiple-arms. It has the characteristics of both configurations: the high stiffness and accuracy of a parallel robot, and a large workspace and compact structure of a serial robot. Two serial-parallel robotic manipulators (a linkage robot and an articulated arm platform robot) were studied in this research on their direct and inverse kinematics, velocity, dynamics, and collision detection.

The direct and inverse kinematics problems were solved using the wrist position of each arm. The inverse velocity problem was solved by the force analysis and the principle of virtual work. Based on the velocity analysis, the singularity problem was researched, and the redundant actuation was analyzed. The dynamic models were established using the Lagrange formulation. An efficient algorithm was introduced to detect link collision. This algorithm can be extended to multiple cooperating robots as well. MATLAB was used to simulate robot motion and to verify various control algorithms.

To demonstrate the mobility and capability of a linkage robot, a prototype was built from off-the-shelf components whenever possible. Rhino robot's controller along with its motors is used to control this linkage robot.

CHAPTER ONE

INTRODUCTION

1.1 Background

Most industrial robots have a serial configuration that consists of links serially jointed together. They have a large workspace with a compact structure. However their positioning errors are significant because of the flexibility, friction, and backlash in each joint.

Parallel manipulators, like those used in flight simulators, have the advantages of high stiffness and accuracy because actuators act in parallel to share a common payload, and because errors are distributive in parallel rather than accumulative in series.

A serial-parallel robot has the characteristics of both configurations: the high stiffness and accuracy of a parallel robot, and a large workspace and compact structure of a serial robot. Notash and Podhorodeski [1994] presented a broad class of three articulated-arm parallel manipulators. These manipulators consist of a mobile platform supported by three arms, each comprised of three revolving joints with a passive spherical joint. Forward displacement solutions were presented for all possible combinations of non-redundant and redundant actuation.

Wang [1991] proposed a 6-bar linkage for a 5 degree-of-freedom (DOF) robot, which can be viewed as a platform robot with two articulated arms. The 6^{th} DOF can be provided with a roll motor at the wrist. The force and motion control of this robot was discussed by You [1992].

For dynamics analysis and torque control, Pang and Shahinpoor [1994] analyzed the inverse dynamics of a 3 DOF hybrid parallel manipulator using Lagrange's method. Baiges and Duffy [1995] derived the explicit dynamic equations of a parallel manipulator using Kane's method. The explicit dynamic equations allowed the designer to examine the effects of different factors such as link dimensions.

A serial-parallel robot has two or more serial arms, and the collision of links is possible. For collision detection of multiple robots, Hurteau and Stewart [1988] proposed a minimum distance algorithm for imminent collision indication in an off-line graphical robot simulation system. The algorithm is based on the minimum distance between pairs of convex polyhedral objects. Chang [1990] used minimum distance functions for planning collision-free motion of two articulated robot arms, and he modeled links as cylinders and spheres. Cao [1993] modeled links the same way and presented a fast

collision detection and avoidance algorithm by using Lagrange multipliers to find the minimum distance between two links.

Amirouche and Jia [1988] presented a method using differential geometry and body vectors to establish norms and planes to formulate the geometrical constraints needed for multi-robot collision avoidance.

1.2 Scope

In Chapter Two, the direct and inverse kinematics of the linkage and platform robots are analyzed to relate the position and orientation of the tool to joint angles. A MATLAB programs were written to verify the kinematics and to demonstrate the motion animation of these robots.

In Chapter Three, a linkage robot with redundant actuation is presented to avoid singularities. Velocity and force analyses based on the position Jacobian matrix of the wrists were performed. This approach is based on serial robot models.

In Chapter Four, velocity and force control algorithms of linkage and platform robots are derived based on parallel manipulators' approach using screw theory and wrenches. Methods based on serial robots and parallel robots should generate the same results. MATLAB programs are coded to verify this hypothesis.

In Chapter Five, the linkage robot's dynamic model is established using the Lagrange formulation to relate the actuator torque to the links' motion. Equations of motion for a three articulated-arm robot are derived similarly.

In Chapter Six, an efficient link collision detection algorithm for linkage and platform robots is presented. Cylinders are used to model links, and line equations with parameters are used to represent center-lines of these cylinders. If the distance between two center-lines is less than the sum of radii of these two cylinders, and the feet of the common normal of the two center-lines are inside the links, collision happens. MATLAB programs are written to simulate the motion and detect link collision.

CHAPTER TWO

KINEMATICS OF LINKAGE AND PLATFORM ROBOTS

A serial robot manipulator can be modeled as a chain of rigid bodies connected by joints. In a parallel robot, the platform is supported by linear actuators in parallel. In a serial-parallel robot, the platform is connected to the base by multiple serial chains. In this chapter, the Denavit-Hartenberg (D-H) coordinate transformation is first reviewed. Then the direct and inverse kinematics of a linkage robot and a three articulated-arm platform robot are solved. Finally, motion simulation programs were written in MATLAB to simulate the straight-line path motion of the linkage and platform robots.

2.1 Denavit-Hartenberg (D-H) Coordinate Transformation

The objective of the D-H coordinate transformation is to systematically assign coordinate frames to links and then to transform position and orientation from the tool frame to the base frame. Coordinate frame $\mathbf{L}_k$ will be attached to the distal end of link *k*. The coordinate $\mathbf{z}^{k-1}$ is defined along joint *k*, and $\mathbf{x}^k$ is defined as the common normal of $\mathbf{z}^{k-1}$ and $\mathbf{z}^k$, as shown in Figure 2.1.

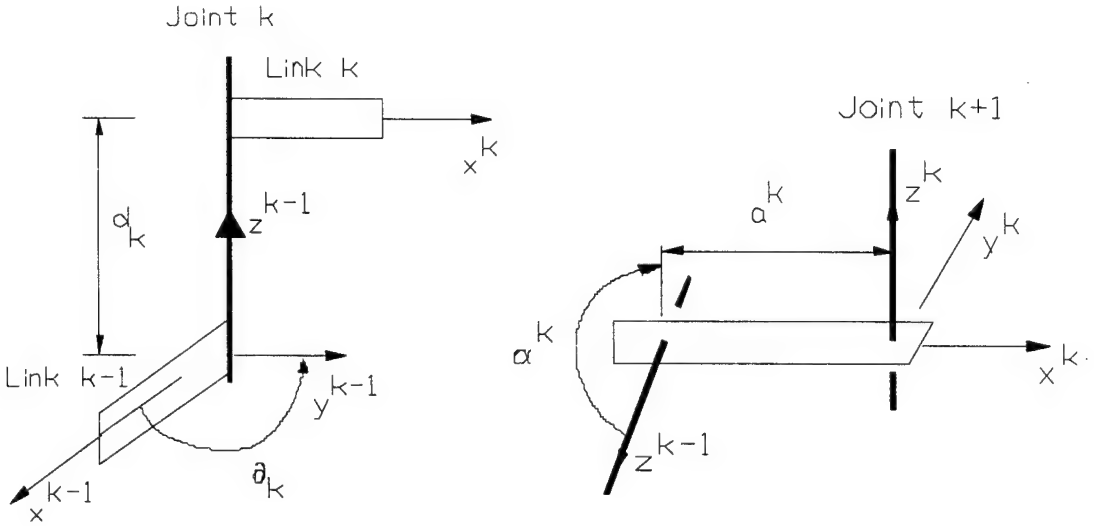

Figure 2.1 **D-H parameters**

Let $\mathbf{p}^k$ and $\mathbf{p}^{k-1}$ be the homogeneous coordinates of a point with respect to the frame $\mathbf{L}_k$ and $\mathbf{L}_{k-1}$ respectively. The coordinate transformation can be represented by

$$\mathbf{p}^{k-1} = \mathbf{T}_{k-1}^{k}\,\mathbf{p}^{k} \tag{2.1}$$

$$\text{where } \mathbf{T}_{k-1}^{k} = \begin{bmatrix} c\theta_k & -c\alpha_k s\theta_k & s\alpha_k s\theta_k & a_k c\theta_k \\ s\theta_k & c\alpha_k c\theta_k & s\alpha_k c\theta_k & a_k s\theta_k \\ 0 & s\alpha_k & c\alpha_k & d_k \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and} \tag{2.2}$$

$s\theta_k$, $c\theta_k$, $s\alpha_k$, $c\alpha_k$ represent $\sin\theta_k$, $\cos\theta_k$, $\sin\alpha_k$, and $\cos\alpha_k$ respectively.

The D-H parameters involved in (2.2) are θ_k, d_k, a_k, and α_k. The joint angle θ_k is the rotation about $\mathbf{z}^{k-1}$ needed to make axis $\mathbf{x}^{k-1}$ parallel to axis $\mathbf{x}^{k}$. The joint distance d_k is the translation along $\mathbf{z}^{k-1}$ needed to make the axis $\mathbf{x}^{k-1}$ intersect $\mathbf{x}^{k}$. The link length a_k is the translation along $\mathbf{x}^{k}$ needed to make axis $\mathbf{z}^{k-1}$ intersect axis $\mathbf{z}^{k}$. The twist angle α_k is the rotation about $\mathbf{x}^{k}$ needed to make axis $\mathbf{z}^{k-1}$ parallel to axis $\mathbf{z}^{k}$.

Four steps are involved in the homogeneous transformation $\mathbf{T}_{k-1}^{k}$, and are summarized in Table 2.1.

Table 2.1 Coordinate transforming from frame k to frame k-1

Steps	Description
1	Rotate $\mathbf{L}_{k-1}$ about $\mathbf{z}^{k-1}$ by θ_k
2	Translate $\mathbf{L}_{k-1}$ along $\mathbf{z}^{k-1}$ by d_k
3	Translate $\mathbf{L}_{k-1}$ along $\mathbf{x}^{k}$ by a_k
4	Rotate $\mathbf{L}_{k-1}$ about $\mathbf{x}^{k}$ by α_k

The transformation matrix (2.2) can be written as $\mathbf{T} = \left[\begin{array}{c|c} \mathbf{R} & \mathbf{p} \\ \hline \mathbf{0} & 1 \end{array}\right]$ (2.3)

where $\mathbf{R}$ and $\mathbf{p}$ are the orientation and position of frame $\mathbf{L}_k$ relative to frame $\mathbf{L}_{k-1}$, $\mathbf{R}$ is a 3×3 rotation matrix, and $\mathbf{p}$ is the translation vector from the origin of the frame $\mathbf{L}_{k-1}$ to that of frame $\mathbf{L}_k$.

2.2 The Kinematics of a Linkage Robot

Figure 2.2 shows the linkage structure of a serial-parallel robot [Wang 1991]. Because the linkage robot has 5 DOF, only five joints need to be active. We can locate the active joints close to the base to reduce the moving inertia. Thus, joints L_1, L_2, L_3, R_1, and R_3 are chosen as being active. By controlling these five joints, the tool's position

as well as pitch and yaw angles can be controlled. The roll angle of the tool can not be controlled, however.

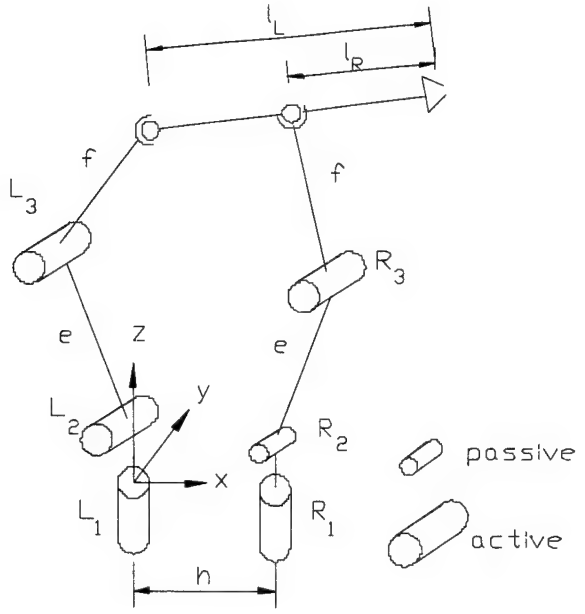

Figure 2.2 A linkage robot

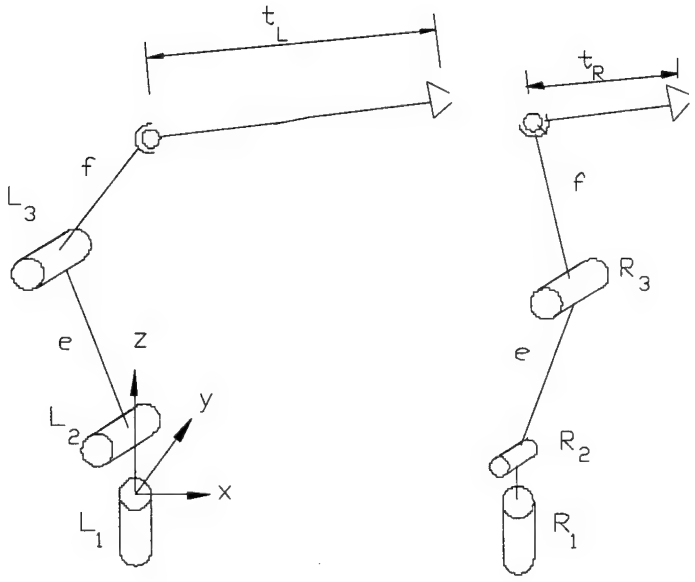

Figure 2.3 The left and right arms of a linkage robot

To simplify the analysis of this linkage robot, we can view it as two connected serial arms as shown in Figure 2.3. The left arm has a 2 DOF shoulder, an elbow and a spherical wrist joint. The right arm has a similar configuration except that its wrist has only 2 DOF.

The coordinate frames of the linkage robot are shown in Figure 2.4. The dashed lines indicate the coincident origins of $\mathbf{L}_0$ and $\mathbf{L}_1$, $\mathbf{L}_2$ and $\mathbf{L}_3$, and $\mathbf{L}_4$ and $\mathbf{L}_5$ respectively. The D-H parameters are listed in Table 2.2.

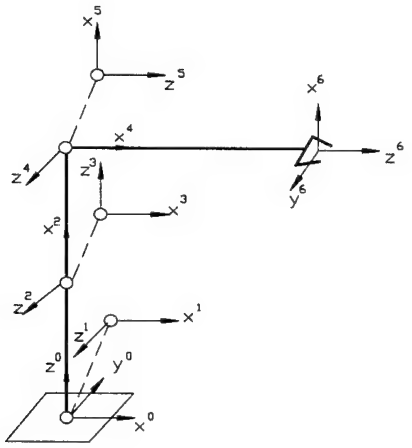

Figure 2.4 Coordinate frames of a linkage robot

Table 2.2 Kinematics parameters of the linkage robot

Axis	θ_i	d_i	a_i	α_i	Home
1	q_1	0	0	$\pi/2$	0
2	q_2	0	a_2	0	$\pi/2$
3	q_3	0	0	$-\pi/2$	$-\pi/2$
4	q_4	d_4	0	$\pi/2$	0
5	q_5	0	0	$-\pi/2$	$\pi/2$
6	q_6	d_6	0	0	0

The transformation matrix of the left arm from the frame *4* to the frame *0* can be expressed as:

$$\mathbf{T}_0^4 = \begin{bmatrix} c_{L1} & 0 & s_{L1} & 0 \\ s_{L1} & 0 & -c_{L1} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{L2} & -s_{L1} & 0 & a_2 c_{L2} \\ s_{L2} & c_{L1} & 0 & a_2 s_{L2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{L3} & 0 & -s_{L3} & 0 \\ s_{L3} & 0 & c_{L3} & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{L4} & 0 & s_{L4} & 0 \\ s_{L4} & 0 & -c_{L4} & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_{L1}c_{L23}c_{L4} - s_{L1}s_{L4} & -c_{L1}s_{L4} & c_{L1}c_{L23}s_{L4} + s_{L1}c_{L4} & a_2 c_{L1}c_{L2} - d_4 c_{L1}s_{L23} \\ s_{L1}c_{L23}c_{L4} + c_{L1}s_{L4} & -s_{L1}s_{L4} & s_{L1}c_{L23}s_{L4} - c_{L1}c_{L4} & a_2 s_{L1}c_{L2} - d_4 s_{L1}s_{L23} \\ s_{L23} & c_{L23} & s_{L23}s_{L4} & a_2 s_{L2} + d_4 c_{L23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

where c_{L1} is $\cos\theta_{L1}$, and s_{L23} is $\sin(\theta_{L2} + \theta_{L3})$.

The inverse kinematics problem is to find joint angles given the position vector **p** and orientation matrix $\mathbf{R} = [\mathbf{n} \quad \mathbf{o} \quad \mathbf{a}]$, where **n**, **o**, **a** are defined as the normal, orienting, and approach vectors, and are the principal axes of the tool coordinate frame. For a 5 DOF robot, only the hand's position **p** and approach vector **a** can be specified.

The position of the left and right wrists can be expressed in terms of **p** and **a** as:

$$\mathbf{p_L} = \mathbf{p} - l_L \mathbf{a} \quad (2.5)$$

$$\mathbf{p_R} = \mathbf{p} - l_R \mathbf{a} \quad (2.6)$$

where l_L and l_R are the distances from the tool to the left and right wrist respectively.

From the **p** in $\mathbf{T}_0^4$ using (2.3), the position of the left wrist can be expressed as:

$$p_{Lx} = a_2 c_{L1} c_{L2} - d_4 c_{L1} s_{L23} \quad (2.7)$$

$$p_{Ly} = a_2 s_{L1} c_{L2} - d_4 s_{L1} s_{L23} \quad (2.8)$$

$$p_{Lz} = a_2 s_{L2} + d_4 c_{L23} \quad (2.9)$$

Solving these equations, the active joint angles can be obtained as:

$$\theta_{L1} = \tan^{-1} \frac{p_{Ly}}{p_{Lx}} \quad (2.10)$$

$$\theta_{L3} = \sin^{-1} \frac{a_2^2 + d_4^2 - p_{Lx}^2 - p_{Ly}^2 - p_{Lz}^2}{2 d_4 a_2} \quad (2.11)$$

$$\theta_{L2} = \tan^{-1} \frac{p_{Lz}(a_2 - d_4 s_{L3}) - d_4 c_{L3}(p_{Lx} c_{L1} + p_{Ly} s_{L1})}{p_{Lz} d_4 c_{L3} + (a_2 - d_4 s_{L3})(p_{Lx} c_{L1} + p_{Ly} s_{L1})} \quad (2.12)$$

The right arm's D-H parameters are similar to those of the left arm, and its joint angles can be solved as:

$$\theta_{R1} = \tan^{-1}\frac{p_{Ry}}{p_{Rx} - h} \tag{2.13}$$

$$\theta_{R3} = \sin^{-1}\frac{a_2^2 + d_4^2 - (p_{Rx} - h)^2 - p_{Ry}^2 - p_{Rz}^2}{2d_4a_2} \tag{2.14}$$

$$\theta_{R2} = \tan^{-1}\frac{p_{Rz}(a_2 - d_4s_{R3}) - d_4c_{R3}[(p_{Rx} - h)c_{R1} + p_{Ry}s_{R1}]}{p_{Rz}d_4c_{R3} + (a_2 - d_4s_{R3})[(p_{Rx} - h)c_{R1} + p_{Ry}s_{R1})]} \tag{2.15}$$

The direct kinematics is to find the hand position and orientation given the active joint positions. The position of the right wrist is a function of θ_{R1}, θ_{R2}, and θ_{R3}, similar to (2.6) to (2.8), where θ_{R1} and θ_{R3} are given, and only θ_{R2} is unknown. The distance between the two wrists is a constant k and can be expressed as a function of the joint angles, including the unknown θ_{R2}, as shown

$$k^2 = \left[p_{Lx} - p_{Rx}(\theta_{R2})\right]^2 + \left[p_{LY} - p_{RY}(\theta_{R2})\right]^2 + \left[p_{LZ} - p_{RZ}(\theta_{R2})\right]^2$$

The angle θ_{R2} can then be obtained by solving this equation. Once θ_{R2} is known, the position of the right wrist can be found using a numerical method. Then the approach vector **a** can be expressed as:

$$\mathbf{a} = \frac{\mathbf{p_R} - \mathbf{p_L}}{|\mathbf{p_R} - \mathbf{p_L}|} \tag{2.16}$$

The position of the end effector can then be obtained in (2.4). In addition to **a**, the other two vectors **n** and **o** of the orientation matrix **R** need to be decided. Unit vector **b**, which is in the direction of the right forearm, can first be decided by the positions of the right elbow and wrist. Once **b** is known, orientation vector **o** can be obtained as $\mathbf{o} = \mathbf{a} \times \mathbf{b}$, and then normal vector **n** can be obtained as $\mathbf{n} = \mathbf{a} \times \mathbf{o}$.

2.3 The Kinematics of a Platform Robot

Because the platform robot has 6 DOF, only six out of all 18 joints need to be active. Therefore, only two joints of each arm are chosen as active, similar to the right arm of the linkage robot. By controlling these six joints, the platform's position and its orientation can be controlled.

The coordinate transformation from a position vector $\mathbf{p_p}$ of the platform coordinates to that of the base coordinates **p** can be expressed as:

$$\mathbf{p} = \mathbf{R}\mathbf{p_p} + \mathbf{p_0} \tag{2.17}$$

The feet on the base and the platform constitute an equilateral triangle respectively, as shown in Figure 2.5. The origin of the global frame is at the base

triangle's center, and that of the tool frame is at the platform triangle's center. The matrix **R** and vector $\mathbf{p}_0$ represent the orientation and translation of the platform vector relative to the base. The feet on the base in the global coordinate frame can be expressed as:

$$b = \left[r_b c_{bi},\ r_b s_{bi},\ 0 \right] \tag{2.18}$$

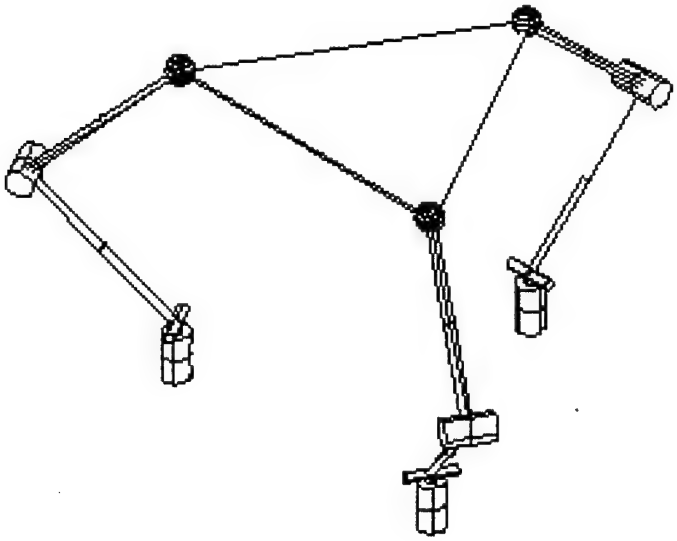

Figure 2.5 A three articulated-arm platform robot

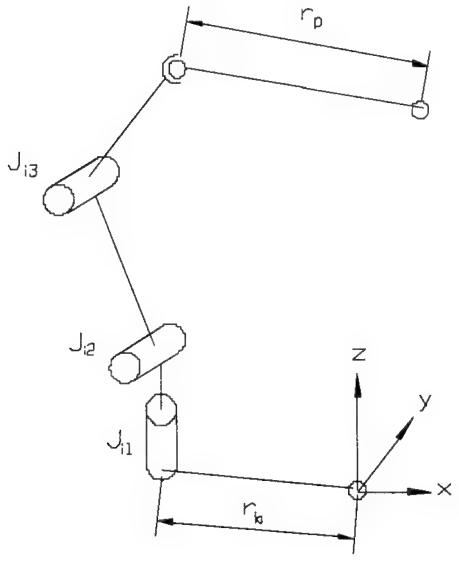

Figure 2.6 One arm of the platform robot

The coordinates of the feet on the platform in the platform coordinate frame can be expressed as:

$$\mathbf{p}_i^p = \left[r_p c_{pi},\ r_p s_{pi},\ 0 \right] \tag{2.19}$$

where r_b and r_p are the radii of the circles of the feet on the base and platform respectively. Using (2.17), the global coordinates $\mathbf{p}_i$ can be obtained as:

$$\mathbf{p}_i = \mathbf{R}\mathbf{p}_i^p + \mathbf{p}_0 \tag{2.20}$$

Each arm of the platform robot can be treated like a serial arm, as shown in figure 2.6. The transformation matrix from the wrist on the platform to the foot on the base can be expressed similar to (2.5) as:

$$^{i}\mathbf{T}_0^4 = \begin{bmatrix} c_{i1}c_{i23}c_{i4} - s_{i1}s_{i4} & -c_{i1}s_{i23} & c_{i1}c_{i23}s_{i4} + s_{i1}c_{i4} & a_2c_{i1}c_{i2} - d_4c_{i1}s_{i23} \\ s_{i1}c_{i23}c_{i4} + c_{i1}s_{i4} & -c_{i1}s_{i23} & s_{i1}c_{i23}s_{i4} - c_{i1}c_{i4} & a_2s_{i1}c_{i2} - d_4s_{i1}s_{i23} \\ s_{i23} & c_{i23} & s_{i23}s_{i4} & a_{i2}s_{i2} + d_4c_{i23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.21}$$

The position vector from a foot on the base to a corresponding wrist on the platform is equal to the position vector in the matrix (2.21) as:

$$\mathbf{p}_i - \mathbf{b}_i = \begin{bmatrix} a_2c_{i1}c_{i2} - d_4c_{i1}s_{i23} \\ a_2s_{i1}c_{i2} - d_4s_{i1}s_{i23} \\ a_2s_2 + d_4c_{i23} \end{bmatrix} \tag{2.22}$$

The joint angles can then be solved similar to (2.10)-(2.12).

2.4 The Prototype Robot

To demonstrate the mobility and capability of this linkage robot, a prototype was built as shown in Figure 2.7. To simplify the construction, off-the-shelf components are used whenever possible. Universal joints are commercially available but not used for wrists because their motion range is limited. An in-line pitch joint along with two roll joints is designed to constitute a spherical wrist. The in-line pitch joint allows the pitch motion of up to 75 degrees, and the stiffness and strength of the tongue and yoke of this joint are preserved by using v-shaped ribs, as shown in Fig. 2.8. The simplicity of this in-line pitch joint's geometry ensures a simple control algorithm.

Selecting a robot controller at an affordable price and with minimum modifications is a challenging issue. Rhino robot's controller along with its motors were selected to control this linkage robot because the Rhino robot is a low-cost educational robot with extensive documentation. Because servo gearmotors are used, the robot is not backdrivable and brakes are not needed.

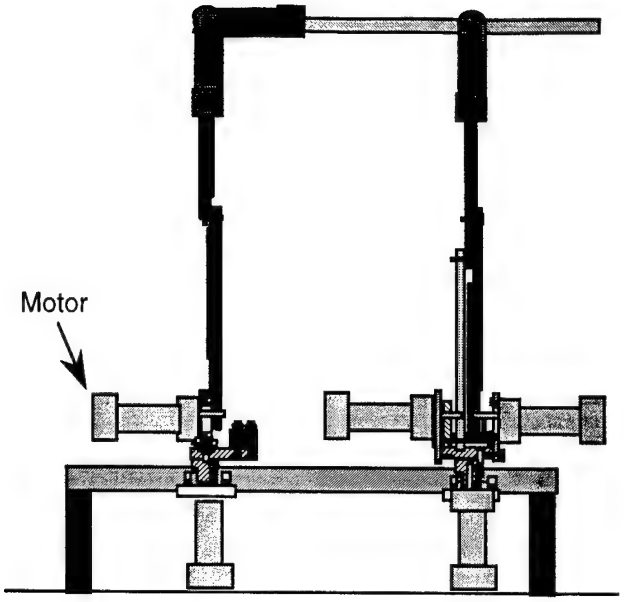

Figure 2.7 A prototype linkage robot

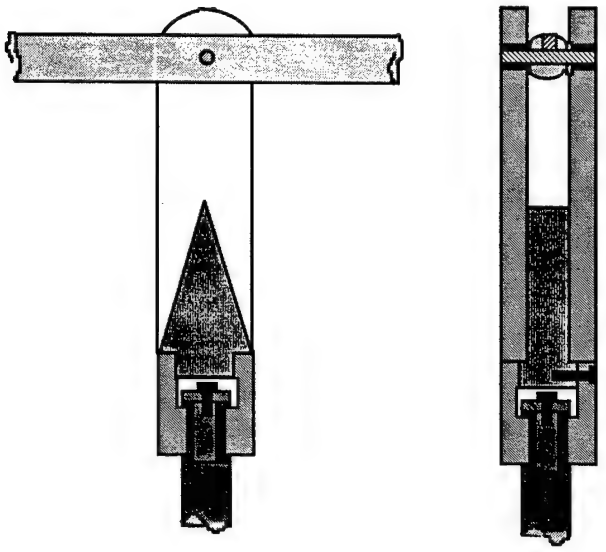

Figure 2.8 The in-line pitch joint

Figure 2.9 shows the Rhino robot in the background sitting on top of the controller. The teach pendant is on the table top and on the left. The prototype robot is on the front, sharing the controller and teaching pendant of the Rhino robot.

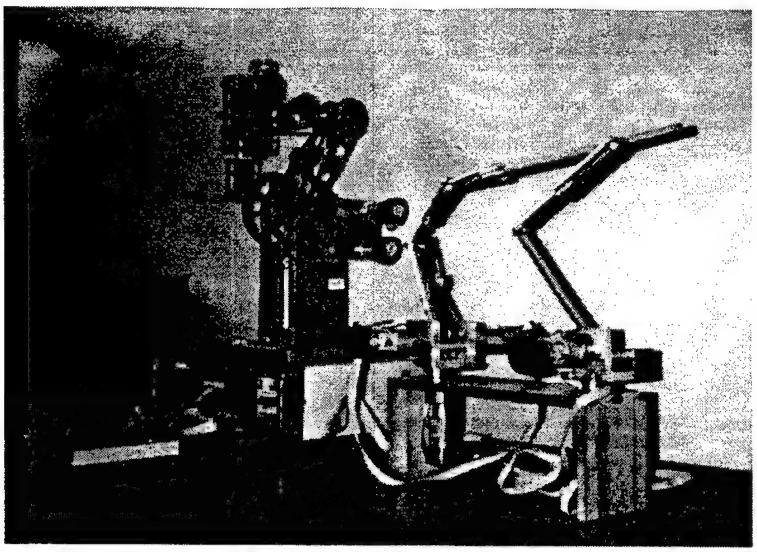

Figure 2.9 The Rhino robot, prototype robot, controller and teach pendant

The controller receives input commands from a teach pendant or an off-line programming environment ROBOTALK [Rhino Robots 1992] on an IBM PC. ROBOTALK can not be used for the linkage robot because it is designed for the Rhino robot's geometry. The motion control programming is feasible because low level control commands are provided [Schilling and White 1990], and programs written in C or BASIC can be downloaded from a PC to the controller through a serial port.

2.5 Motion Simulation Programs

MATLAB programs were written to simulate the straight-line trajectory motion of the linkage and platform robots. The user can specify the starting and ending position, and the orientation of the tool. The simulation programs use knot points and point-to-point control to approximate a straight line motion.

Figure 2.10 shows a screen of the platform robot's simulation. The user can also change the eye position to view the simulation in different directions. The "*start*" button is to start the simulation, and the "*info*" gives the help information.

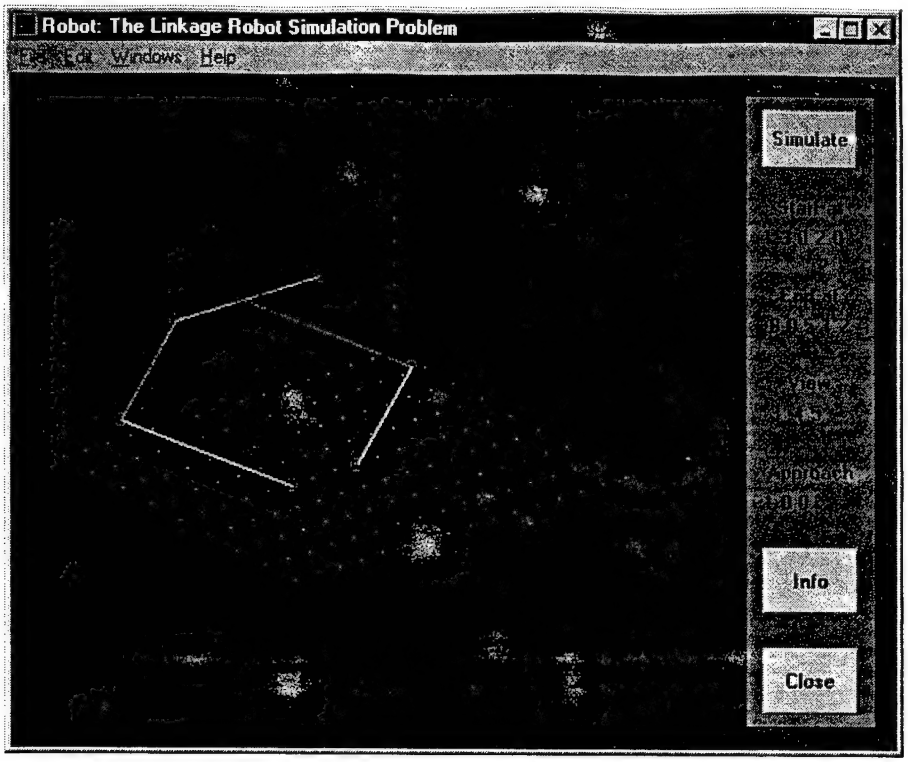

Figure 2.10 An example of the motion simulation program

CHAPTER THREE

MOTION AND FORCE CONTROL WITH REDUNDANT ACTUATION

As discussed in Chapter Two, five motors are needed to control the 5 DOF motion of the linkage robot. These five motors should be located close to the base to reduce moving inertia. If all major axes except the right elbow are active, these five motors are insufficient to deliver the five intended freedoms of the linkage robot in some configurations, and this can be illustrated with a force analysis.

Figure 3.1 shows a free-body diagram of the robot. Since the two wrists are passive, only reaction forces are present to balance the external load. Force f_L at the left wrist has three components controlled by joint torques in the left arm, τ_{L1}, τ_{L2}, and τ_{L3}. Each force is along an axis reciprocal to the other two joint axes in the same arm [Notash and Podhorodeski 1994]. f_{L1} is provided by τ_{L1}, and is in the direction parallel to the second and the third joints. Therefore f_{L2} is provided by τ_{L2}, and is the force in the direction of link L3. f_{L3} is provided by τ_{L3}, and is the force in the direction from the left shoulder to the left wrist. Force f_R at the right wrist has two components, f_{R1} and f_{R2}, defined similar to those at the left wrist.

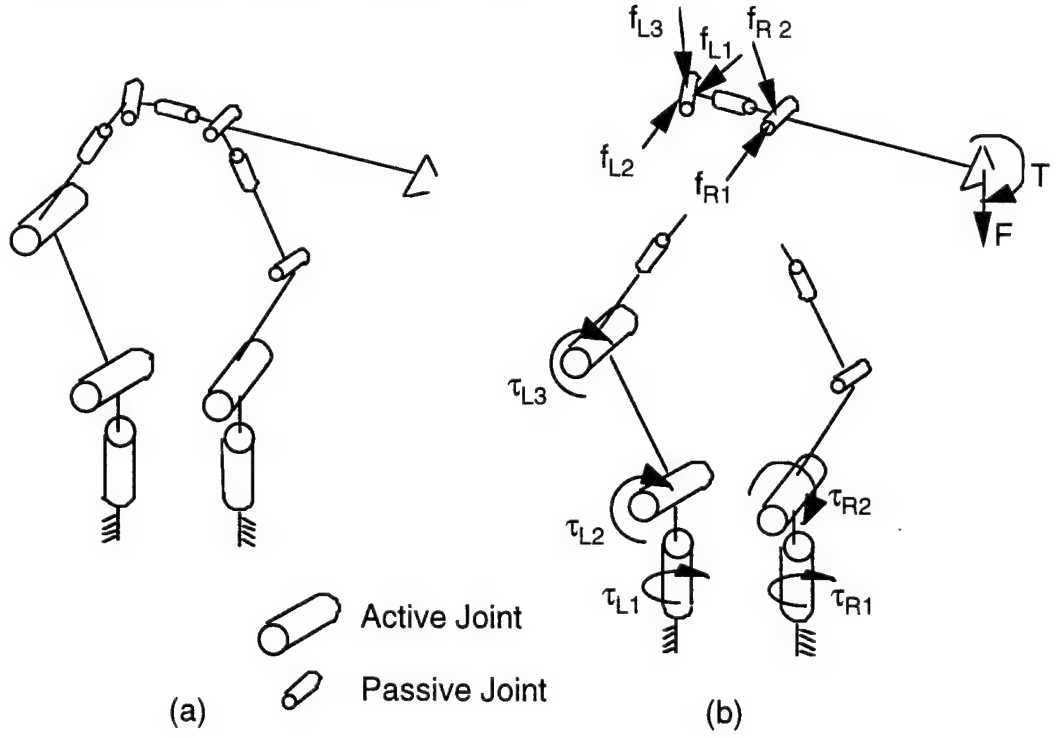

Figure 3.1 The free body diagram of a linkage robot

In Figure 3.2, the right forearm is at a horizontal position. The two reaction forces at the right wrist will be on a horizontal plane, and therefore the coupler-arm can not be properly supported if an external moment is in the vertical plane. In general, since the right wrist's force has only two components, a torque vector normal to the plane defined by these two forces can not be supported. Therefore, a linkage robot with five motors can have only four degrees of freedom. In order to become a five DOF robot, the right elbow must also be actuated as shown in Figure 3.3.

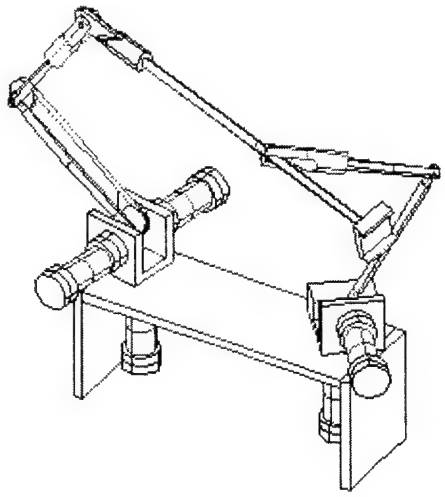

Figure 3.2 A linkage robot at an immobile position

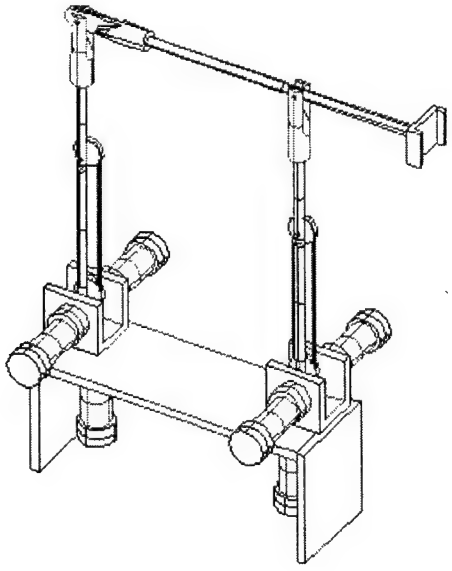

Figure 3.3 A linkage robot with redundant actuation

3.1 Rates Control Algorithm

The velocity analysis is to relate the joint rate to the hand velocity. We can first relate the left wrist's velocity $\mathbf{v}_{Lw}$ to the hand's linear and angular velocity $\mathbf{v}$ and $\mathbf{w}$ as:

$$\mathbf{v}_{Lw} = \mathbf{v} - \mathbf{w} \times l_L \mathbf{a} \tag{3.1}$$

As in (2.7) to (2.9), the left wrist's position can be expressed as functions of joint angles θ_{L1}, θ_{L2}, and θ_{L3}.

$$\begin{aligned} p_{Lx} &= a_2 c_{L1} c_{L2} - d_3 c_{L1} s_{L23} \\ p_{Ly} &= a_2 s_{L1} c_{L2} - d_3 s_{L1} s_{L23} \\ p_{Lz} &= a_2 s_{L2} + l_3 c_{L23} \end{aligned} \tag{3.2}$$

Taking time derivatives of (3.2), we have:

$$\mathbf{v}_{Lw} = \mathbf{J}_{Lp} \dot{\mathbf{q}}_{Lp} \tag{3.3}$$

where $\dot{\mathbf{q}}_{Lp} = \left[\dot{\theta}_{L1}, \dot{\theta}_{L2}, \dot{\theta}_{L3}\right]^T$, and $\mathbf{J}_{Lp}$ is a 3x3 position Jacobian matrix.

$$\begin{aligned} \mathbf{J}_{Lp} &= \begin{bmatrix} \frac{\partial p_{Lx}}{\partial \theta_{L1}} & \frac{\partial p_{Lx}}{\partial \theta_{L2}} & \frac{\partial p_{Lx}}{\partial \theta_{L3}} \\ \frac{\partial p_{Ly}}{\partial \theta_{L1}} & \frac{\partial p_{Ly}}{\partial \theta_{L2}} & \frac{\partial p_{Ly}}{\partial \theta_{L3}} \\ \frac{\partial p_{Lz}}{\partial \theta_{L1}} & \frac{\partial p_{Lz}}{\partial \theta_{L2}} & \frac{\partial p_{Lz}}{\partial \theta_{L3}} \end{bmatrix} \\ &= \begin{bmatrix} s_{L1}(a_2 c_{L2} - d_4 s_{L23}) & -c_{L1}(a_2 s_{L2} + d_4 c_{L23}) & -d_4 c_{L1} c_{L23} \\ c_{L1}(a_2 c_{L2} - d_4 s_{L23}) & -s_{L1}(a_2 s_{L2} + d_4 c_{L23}) & -d_4 s_{L1} c_{L23} \\ 0 & a_2 c_{L2} - d_4 s_{L23} & -d_4 c_{L23} \end{bmatrix} \end{aligned} \tag{3.4}$$

Inverting (3.3), the joint rates can be related to the wrist velocity as:

$$\dot{\mathbf{q}}_{Lp} = \mathbf{J}_{Lp}^{-1} \mathbf{v}_{Lw} \tag{3.5}$$

Similarly, we can relate the right joint rates to the right wrist's velocity as:

$$\dot{\mathbf{q}}_{Rp} = \mathbf{J}_{Rp}^{-1} \mathbf{v}_{Rw} \tag{3.6}$$

Combining (3.5) and (3.6), we have:

$$\dot{\mathbf{q}} = \mathbf{J}_p^{-1} \mathbf{v}_w \tag{3.7}$$

where $\dot{\mathbf{q}} = \left[\dot{\theta}_{L1}, \dot{\theta}_{L2}, \dot{\theta}_{L3}, \dot{\theta}_{R1}, \dot{\theta}_{R2}, \dot{\theta}_{R3}\right]^T$, $\mathbf{v}_w = \begin{bmatrix} \mathbf{v}_{Lw} \\ \mathbf{v}_{Rw} \end{bmatrix}$, and $\mathbf{J}_p = \begin{bmatrix} \mathbf{J}_{Lp} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{Rp} \end{bmatrix}$

$$\mathbf{J_{ip}} = \begin{bmatrix} -s_{i1}(l_2c_{i2} + l_3c_{i23}) & -c_{i1}(l_2s_{i2} + l_3s_{i23}) & -l_3c_{i1}s_{i23} \\ c_{i1}(l_2c_{i2} + l_3c_{i23}) & -s_{i1}(l_2s_{i2} + l_3s_{i23}) & -l_3s_{i1}s_{i23} \\ 0 & l_2c_{i2} + l_3c_{i23} & -l_3c_{i23} \end{bmatrix}, \quad i=L,R$$

We can rewrite (3.1) to replace the cross product by a matrix-vector multiplication as:

$$\mathbf{v_{Lw}} = \mathbf{v} + l_L\mathbf{Aw} \tag{3.7}$$

where $\mathbf{A} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$, and a_x, a_y, a_z are components of vector $\mathbf{a}$.

Similar to (3.7), we can have:

$$\mathbf{v_{Rw}} = \mathbf{v} + l_R\mathbf{Aw} \tag{3.8}$$

Combining (3.7) and (3.8), we have:

$$\mathbf{v_w} = \mathbf{B_1}\begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} \tag{3.9}$$

where $\mathbf{B_1} = \begin{bmatrix} \mathbf{I} & d_L\mathbf{A} \\ \mathbf{I} & d_R\mathbf{A} \end{bmatrix}$ (3.10)

Substituting (3.9) into (3.6), we have:

$$\dot{\mathbf{q}} = \mathbf{J_p^{-1}B_1}\begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} \tag{3.11}$$

Matrix $\mathbf{B_1}$ is singular because in (3.9), the wrist velocity is independent of the hand's roll velocity ω_a. Therefore only five components of the hand velocity $\begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}$ are mapped to a six-component wrist velocity $\mathbf{v_w}$, and $\mathbf{B_1}$ is not full rank. To exclude ω_a in the velocity analysis, we can relate $\mathbf{w}$ the angular velocity in the tool coordinate frame to that in the Cartesian coordinate frame $\mathbf{w_t} = [\omega_n, \omega_o, \omega_a]^T$ as:

$$\mathbf{w} = \mathbf{Rw_t} \tag{3.12}$$

where $\mathbf{w} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$, and $\mathbf{w_t} = \begin{bmatrix} \omega_n \\ \omega_o \\ \omega_a \end{bmatrix}$

Substituting (3.12) into (3.7) and (3.8), we can have:

$$\mathbf{v}_w = \mathbf{B}_2 \begin{bmatrix} \mathbf{v} \\ \mathbf{w}_t \end{bmatrix} \tag{3.13}$$

where $\mathbf{B}_2 = \begin{bmatrix} \mathbf{I} & l_L\mathbf{AR} \\ \mathbf{I} & l_R\mathbf{AR} \end{bmatrix}$

The last column of $\mathbf{B}_2$ is a zero vector, and that is why ω_a is not relevant in (3.13). We can then rewrite (3.13) as:

$$\mathbf{v}_w = \mathbf{B} \begin{bmatrix} \mathbf{v} \\ \mathbf{w}_t \end{bmatrix}_5 \tag{3.14}$$

where $\begin{bmatrix} \mathbf{v} \\ \mathbf{w}_t \end{bmatrix}_5$ is the five-component hand velocity, obtained by eliminating ω_a, the last component of $\begin{bmatrix} \mathbf{v} \\ \mathbf{w}_t \end{bmatrix}$. $\mathbf{B}$ is a 6×5 matrix, obtained by eliminating the last column (the zero vector) of $\mathbf{B}_2$. $\mathbf{B}$ can then be expressed as:

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & l_L o_x & l_L n_x \\ 0 & 1 & 0 & l_L o_y & l_L n_y \\ 0 & 0 & 1 & l_L o_z & l_L n_z \\ 1 & 0 & 0 & l_R o_x & l_R n_x \\ 0 & 1 & 0 & l_R o_y & l_R n_y \\ 0 & 0 & 1 & l_R o_z & l_R n_z \end{bmatrix}$$

Substituting (3.14) into (3.6), we have the resolved rate motion control algorithm as:

$$\dot{\mathbf{q}} = \mathbf{J}_p^{-1}\mathbf{B} \begin{bmatrix} \mathbf{v} \\ \mathbf{w}_t \end{bmatrix}_5 \tag{3.15}$$

3.2 Force Control Algorithm

The force analysis is to relate the joint torque $\mathbf{t} = [\tau_{L1}, \tau_{L2}, \tau_{L3}, \tau_{R1}, \tau_{R2}, \tau_{R3}]$ to the external load $\begin{bmatrix} \mathbf{F} \\ \mathbf{T} \end{bmatrix} = [F_x, F_y, F_z, T_x, T_y, T_z]^T$. Using the principle of virtual work [Paul 1981], we can relate the joint torque to the wrist force by the same $\mathbf{J}_p$ defined in (3.5) as:

$$\mathbf{t} = \mathbf{J}_p{}^T \mathbf{f}_w \tag{3.16}$$

where $\mathbf{t}$ is the joint torque, and $\mathbf{f}_w = \begin{bmatrix} \mathbf{f}_{Lw} \\ \mathbf{f}_{Rw} \end{bmatrix}$ is the wrist force in the Cartesian coordinates, not $f_{L1}, f_{L2}, \ldots, f_{R3}$ defined in the previous section. To relate the external load $\begin{bmatrix} \mathbf{F} \\ \mathbf{T} \end{bmatrix}$ to the wrist force, we can perform a force and moment equilibrium as:

$$\mathbf{F} = \sum \mathbf{f}_w$$

$$\mathbf{T} = \sum t_L \mathbf{a} \times \mathbf{f}_{Lw} + \sum t_R \mathbf{a} \times \mathbf{f}_{Rw}$$

Combining these two equations, we have:

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{T} \end{bmatrix} = \mathbf{B}_1^T \mathbf{f}_w \tag{3.17}$$

where $\mathbf{B}_1$ is defined in (3.10). Since torque about the coupler arm T_a can not be supported by $\mathbf{f}_w$, we can use a five DOF payload $\begin{bmatrix} \mathbf{F} \\ \mathbf{T}_t \end{bmatrix}_5$, where the subscript $\mathbf{t}$ indicates the torques in the tool coordinate frame, and $\mathbf{F}$ has 3 components expressed in the Cartesian coordinate frame, and $\mathbf{T}_t = \begin{bmatrix} T_n \\ T_o \end{bmatrix}$. Equation (3.17) can then be rewritten as:

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{T}_t \end{bmatrix}_5 = \mathbf{B}^T \mathbf{f}_w \tag{3.18}$$

Because $\mathbf{B}$ is not a square matrix, it can not be inverted. The pseudo-inverse [Nakamura 1991] will be used to relate the wrist force to the external load.

$$\mathbf{f}_w = \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1} \begin{bmatrix} \mathbf{F} \\ \mathbf{T}_t \end{bmatrix}_5 \tag{3.19}$$

Substituting (3.19) into (3.17), we obtain the resolved force control as:

$$\mathbf{t} = \mathbf{J}_p{}^T \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1} \begin{bmatrix} \mathbf{F} \\ \mathbf{T}_t \end{bmatrix}_5 \tag{3.20}$$

Notice that the pseudo-inverse solution produces a minimum norm least-squares solution [Nakamura 1991] of the wrist force in (3.19), and in turn produces a minimum norm least-squares solution of the joint torque in (3.20).

CHAPTER FOUR

FORCE AND MOTION CONTROL BASED ON WRENCHES

The resolved force control algorithm based on wrenches is derived in this chapter for serial-parallel robots, the three articulated-arm platform robot, and the linkage robot. The resolved rate control algorithm is derived from the force control algorithm based on the principle of virtual work.

Wang [1994] used the wrenches of screw theory [Hunt 1978] to derive a force control algorithm for a full parallel robot. This approach is dual to the velocity control approach discussed in Fichter [1986]. The force control approach is the better approach for serial-parallel robots because redundant information about passive joint rates is not required. MATLAB is used to verify these control algorithms. Graphic animation with stick models of robots is also accomplished with these programs.

4.1 A Three Articulated-arm Platform Robot

As discussed in Chapter Two, only the base and elbow joints of each arm of the platform robot need to be active. Each active joint controls a corresponding wrist force on the platform.

Figure 4.1 shows a free-body diagram of the platform robot. Each wrist force has two components $\mathbf{f}_{i1}$ and $\mathbf{f}_{i3}$, where i indicates the wrist number or the arm number, and the second subscript indicates the active joint. Each $\mathbf{f}_{ij}$ passes through the wrist i and is along a line reciprocal to the other two major joint axes in the same arm, as discussed in Chapter Two.

The force analysis is to relate the joint torque $\mathbf{t} = [\tau_{11}, \tau_{13}; \tau_{21}, \tau_{23}; \tau_{31}, \tau_{33}]$ to the external load $\begin{bmatrix} \mathbf{F} \\ \mathbf{T} \end{bmatrix} = [F_x, F_y, F_z, T_x, T_y, T_z]^T$. The first step is to relate the external force $\mathbf{F}$ to the wrist force $\mathbf{f} = [f_{11}, f_{13}; f_{21}, f_{23}; f_{31}, f_{33}]$ using the force equilibrium:

$$\mathbf{F} = \sum_{i=1}^{3} \sum_{j=1,3} \mathbf{f}_{ij} = \sum_{i=1}^{3} \sum_{j=1,3} \mathbf{u}_{ij} f_{ij} \tag{4.1}$$

where $\mathbf{u}_{ij}$ is the unit vector in the direction of the wrist force. $\mathbf{u}_{i1} = \begin{bmatrix} s_{i1} \\ -c_{i1} \\ 0 \end{bmatrix}$, and

$\mathbf{u_{i3}} = \frac{1}{g_i}\begin{bmatrix} c_{i1}(a_2 c_{i2} - d_4 s_{i23}) \\ s_{i1}(a_2 c_{i2} - d_4 s_{i23}) \\ a_2 s_{i2} + d_4 c_{i23} \end{bmatrix}$. If joint $i2$ is active, $\mathbf{u}_{i2} = \begin{bmatrix} -c_{i1}s_{i23} \\ -s_{i1}s_{i23} \\ c_{i23} \end{bmatrix}$, and $g_i = \sqrt{a_2^2 + d_4^2 - 2a_2 d_4 s_{i3}}$ is the distance from a respective shoulder i to its wrist.

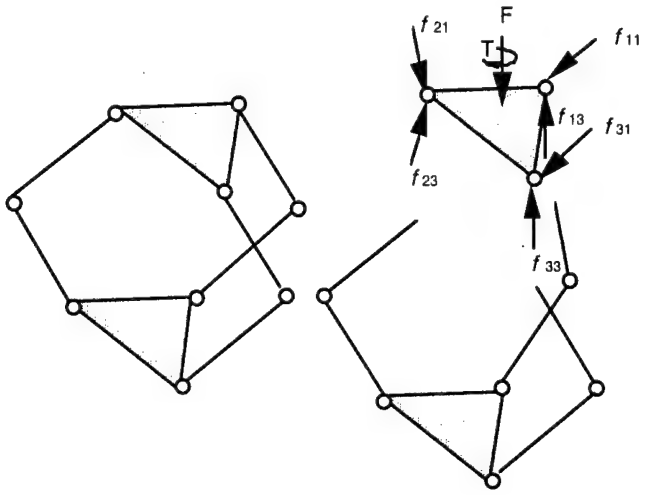

Figure 4.1 Free body diagram of a three articulated-arm platform robot

The external torque $\mathbf{T}$ can be related to the wrist force using moment equilibrium:

$$\mathbf{T} = \sum_{i=1}^{3} \sum_{j=1,3} \mathbf{l}_i \times \mathbf{f}_{ij} \tag{4.2}$$

where $\mathbf{l}_i$ is the distance from each wrist to the center of the platform. Combining (4.1) and (4.2), we have:

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{T} \end{bmatrix} = \mathbf{V}\mathbf{f} \tag{4.3}$$

Each column of $\mathbf{V}$ is a wrench, representing the wrist force in the screw coordinates as $\begin{bmatrix} \mathbf{u}_{ij} \\ \mathbf{l}_i \times \mathbf{u}_{ij} \end{bmatrix}$ (4.4)

Inverting (4.3), we can solve for the wrist force as:

$$\mathbf{f} = \mathbf{V}^{-1}\begin{bmatrix} \mathbf{F} \\ \mathbf{T} \end{bmatrix} \tag{4.5}$$

Joint torque τ_{ij} can be related to wrist force f_{ij} by h_{ij}, the distance from joint ij to the line containing the wrist force $\mathbf{f}_{ij}$ as shown in Figure 4.2.

$$\tau_{ij} = h_{ij} f_{ij} \tag{4.6}$$

where $h_{i1} = |a_2 c_{i2} - d_4 s_{i23}|$ and $h_{i3} = \dfrac{|a_2 d_4 c_{i3}|}{g_i}$. (4.7)

When joint $i2$ is active, $h_{i2} = |a_2 c_{i3}|$. From (4.5) and (4.6) we can get the resolved force control algorithm relating joint torque to the external load as:

$$\mathbf{t} = \mathbf{S}^{-1} \begin{bmatrix} \mathbf{F} \\ \mathbf{T} \end{bmatrix} \tag{4.8}$$

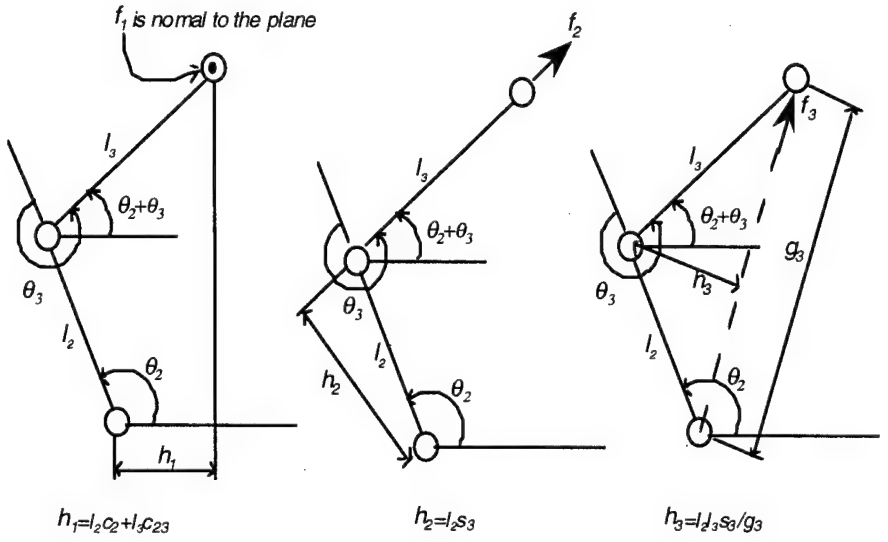

Figure 4.2 Distance between an active joint and a corresponding wrist force

Each column of $\mathbf{S}$ can be represented by that of $\mathbf{V}$ divided by h_{ij} as :

$$\begin{bmatrix} \mathbf{u}_{ij} / h_{ij} \\ (\mathbf{l}_i \times \mathbf{u}_{ij}) / h_{ij} \end{bmatrix} \tag{4.9}$$

Using the principle of virtual work [Paul 1981], we can get the resolved force control algorithm relating the joint rates $\dot{\mathbf{q}} = [\dot{\theta}_{11}, \dot{\theta}_{13}; \dot{\theta}_{21}, \dot{\theta}_{23}; \dot{\theta}_{31}, \dot{\theta}_{33}]^T$ to the hand velocity $[\mathbf{v}, \mathbf{w}]^T$ as:

$$\dot{\mathbf{q}} = \mathbf{S}^{\mathrm{T}} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} \tag{4.10}$$

The resolved rate control algorithm can also be derived using wrist velocity $\mathbf{v}_{iw}$. We can first relate the wrist velocity to the hand velocity as:

$$\mathbf{v}_{\mathbf{iw}} = \mathbf{v} - \mathbf{w} \times \mathbf{l}_{\mathbf{i}} \tag{4.11}$$

The wrist velocity can be related to the rate of major joints in the same arm $\dot{\mathbf{q}}_i = [\dot{\theta}_{i1}, \dot{\theta}_{i2}, \dot{\theta}_{i3}]^T$ by $\mathbf{J}_{\mathbf{ip}}$, the position Jacobian matrix, as

$$\mathbf{v}_{iw} = \mathbf{J}_{ip} \dot{\mathbf{q}}_i \tag{4.12}$$

where $$\mathbf{J}_{ip} = \begin{bmatrix} s_{i1}(-a_2 c_{i2} + d_4 s_{i23}) & -c_{i1}(a_2 s_{i2} + d_4 c_{i23}) & -d_4 c_{i1} c_{i23} \\ c_{i1}(a_2 c_{i2} - d_4 s_{i23}) & -s_{i1}(a_2 s_{i2} + d_4 c_{i23}) & -d_4 s_{i1} c_{i23} \\ 0 & a_2 c_{i2} - d_4 s_{i23} & -d_4 c_{i23} \end{bmatrix} \tag{4.13}$$

Inverting (4.11), the joint rates can be related to the wrist velocity as:

$$\dot{\mathbf{q}}_i = \mathbf{J}_{ip}^{-1} \mathbf{v}_{iw} \tag{4.14}$$

At a singularity (or uncertainty) configuration of a parallel robot, the external load cannot be balanced by the robot, and the robot will collapse ([Gosselin and Angeles 1990], [Hunt 1978 and 1983], [Merlet 1989], [Mohammadi 1993]). Thus $\mathbf{V}$ cannot be inverted at this configuration, and (4.8) will not yield a solution.

The robot is in a singular position when all three $\mathbf{f}_{i1}$ are coplanar and parallel. This will happen if the triangles of the base and platform are of the same size, and when the platform is horizontal. To avoid this singularity we can have different sizes of the platform and base.

The other possible singularity is that all three $\mathbf{f}_{i1}$ are coplanar and concurrent. This can only happen when the size of its platform and base are identical and the feet of the platform are right on top of those of the base. This singularity can also be avoided with different sizes of the platform and base.

Four wrenches $\mathbf{f}_{ij}$ can be dependent and cause singularity. This is possible if a wrist force $\mathbf{f}_{ij}$ is coplanar with the other three. If joint $i2$ (a shoulder) is active, this can happen when the forearm is coplanar with a horizontal platform, and therefore coplanar with three $\mathbf{f}_{i1}$. This is why we need to actuate the elbow joints instead of the shoulder joints. If the elbow is active, $\mathbf{f}_{i3}$ is coplanar with the platform only when the platform and the arm are both on the ground. This singularity can be prevented by using joint limits on the shoulder.

When h_{ij} is zero, the corresponding column in $\mathbf{S}$ is not defined, and (4.9) cannot be used to find joint torque. This singularity is a serial singularity: the velocity at the corresponding wrist loses one freedom ([Waldron 1985], [Wang and Waldron 1987]). h_{i1} is zero when the wrist is on top of the shoulder, and this is a shoulder singularity of a serial robot. h_{i2} or h_{i3} is zero when the arm is fully stretched ($s_i = 0$), and this is an elbow singularity of a serial robot. This serial singularity will not cause problems in the resolved force control (4.5). However because the distance is zero, the joint torque is zero. This means the wrist forces are not supported by joint motors but supported by the structure or frame.

4.2 A Linkage Robot

If one of the arms in the platform robot is passive, the robot cannot have the freedom of rotation or torque about the line connecting the remaining two wrists. If we extend the link connecting the two wrists as a coupler link, as shown in Fig. 2.3, we have the five DOF linkage robot. The sixth freedom, roll motion, can be accomplished with a roll motor attached at the end of the coupler-arm. In the left arm, all three major joints are active; in the right arm, the elbow and wrist are active. Figure 4.3 shows the free-body diagram of the robot. At the left wrist, there are three force components $\mathbf{f}_{11}$, $\mathbf{f}_{12}$ and $\mathbf{f}_{13}$; at the right wrist, there are two force components $\mathbf{f}_{21}$ and $\mathbf{f}_{23}$.

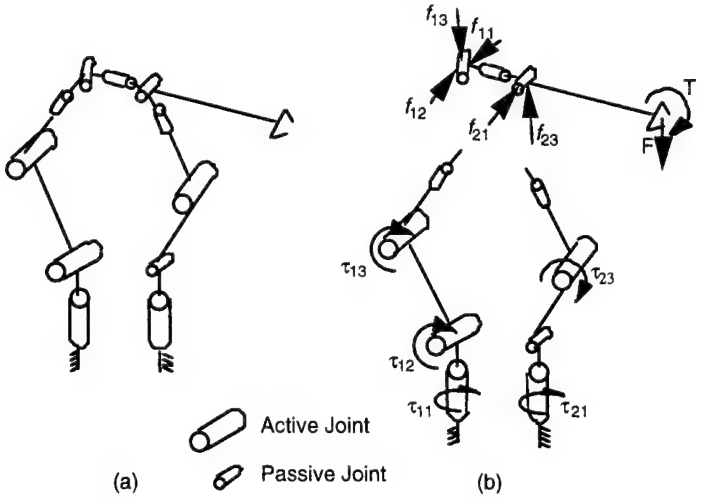

Figure 4.3 The free body diagram of a linkage robot

The force analysis is to relate joint torques to the external load. The first step is to relate the external force $\mathbf{F}$ to wrist force $\mathbf{f}_5 = [f_{11}, f_{12}, f_{13}; f_{21}, f_{23}]$ using the force

equilibrium.

$$\mathbf{F} = \sum_{i=1}^{2} \sum_{j=1,3,2} \mathbf{f}_{ij} = \sum_{i=1}^{2} \sum_{j=1,3,2} \mathbf{q}_{ij} f_{ij} \tag{4.15}$$

The external torque $\mathbf{T}$ can be related to the wrist force as:

$$\mathbf{T} = \sum_{i=1}^{2} \sum_{j=1,3,2} l_i \left(\mathbf{a} \times \mathbf{f}_{ij} \right) \tag{4.16}$$

where l_i is the distance from each wrist to the hand, and $\mathbf{a}$ is the approach vector, in the direction of the coupler-arm. The normal and orienting components of the torque $\mathbf{T}$ can be obtained by dot products as:

$$T_n = \mathbf{n} \cdot \mathbf{T} \text{ and } T_o = \mathbf{o} \cdot \mathbf{T} \tag{4.17}$$

Combining (4.15), (4.16) and (4.17), we can relate $\begin{bmatrix} \mathbf{F} \\ \mathbf{T}_t \end{bmatrix}_5 = [F_x, F_y, F_z; T_n, T_o]^T$, the 5 DOF external load, to $\mathbf{f}_5$ as:

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{T}_t \end{bmatrix}_5 = \mathbf{V}_5 \mathbf{f}_5 \tag{4.18}$$

Each column of $\mathbf{V}_5$ can be obtained by substituting (4.17) into (4.16) as $\begin{bmatrix} \mathbf{u}_{ij} \\ l_i \mathbf{n} \cdot (\mathbf{a} \times \mathbf{u}_{ij}) \\ l_i \mathbf{o} \cdot (\mathbf{a} \times \mathbf{u}_{ij}) \end{bmatrix}$

Inverting (4.18), we can get: $\mathbf{f}_5 = \mathbf{V}_5^{-1} \begin{bmatrix} \mathbf{F} \\ \mathbf{T}_t \end{bmatrix}_5$ (4.19)

Similar to (4.8), we can relate joint torques $\mathbf{t}_5 = [\tau_{11}, \tau_{12}, \tau_{13}; \tau_{21}, \tau_{23}]^T$ to the external load as: $\mathbf{t}_5 = \mathbf{S}_5^{-1} \begin{bmatrix} \mathbf{F} \\ \mathbf{T}_t \end{bmatrix}_5$ (4.20)

Each column of $\mathbf{S}_5$ can be represented by $\begin{bmatrix} \mathbf{u}_{ij} / h_{ij} \\ l_i \mathbf{n} \cdot (\mathbf{a} \times \mathbf{u}_{ij}) / h_{ij} \\ l_i \mathbf{o} \cdot (\mathbf{a} \times \mathbf{u}_{ij}) / h_{ij} \end{bmatrix}$ (4.21)

Using the principle of virtual work, we can relate $\dot{\mathbf{q}}_5 = [\dot{\theta}_{11}, \dot{\theta}_{12}, \dot{\theta}_{13}; \dot{\theta}_{21}, \dot{\theta}_{23}]^T$ the joint rate to the hand velocity as:

$$\dot{\mathbf{q}}_5 = \mathbf{S}_5^T \begin{bmatrix} \mathbf{v} \\ \mathbf{w}_t \end{bmatrix}_5 \tag{4.22}$$

where $\begin{bmatrix} \mathbf{v} \\ \mathbf{w}_t \end{bmatrix}_5 = [v_x, v_y, v_z; \omega_n, \omega_o]$ is the five-component hand velocity, obtained by eliminating ω_o, the last component of $\mathbf{w}_t$.

The velocity analysis of this linkage robot can also be obtained by relating the wrist's velocity $\mathbf{v}_{iw}$ to the hand's linear and angular velocity $\mathbf{v}$ and $\mathbf{w}$ as (3.1). The joint rates can be related to the wrist velocity as in (4.14). This way we will get joint rates of all major axes of both arms which include the passive joint rate $\dot{\theta}_{22}$.

The five wrist forces $\mathbf{f}_{ij}$ belong to a five-system of screws [Hunt 1978]. The screw system will be degenerated to a four-system when the left arm is fully stretched. This is because $\mathbf{f}_{i2}$ and $\mathbf{f}_{i3}$ are collinear, and the robot's freedom is reduced by one. In the force field, the freedom lost is the torque in the plane defined by the left arm and the coupler-arm. In the velocity field, the freedom lost is the velocity along the direction in the left arm. $\mathbf{J}_p$ is then singular and the rate motion control algorithm fails.

If the right shoulder is active and the right forearm is at a horizontal position, as shown in Fig. 4.4, $\mathbf{f}_{21}$, $\mathbf{f}_{21}$, and $\mathbf{f}_{31}$ will be on a horizontal plane, and the remaining two forces $\mathbf{f}_{12}$ and $\mathbf{f}_{13}$ cannot support an external moment in the normal direction. This is why the elbow, instead of the shoulder, is active. The singularity caused by the elbow joint can be avoided by joint limits, the same way as in the platform robot.

4.3 Forward Velocity Analysis

The forward velocity analysis of the linkage robot is to find the hand velocity given the joint rates. This can be accomplished by relating the velocity of the two wrists. The right wrists' velocity can be obtained similar to the left wrist's velocity in (3.1) as:

$$\mathbf{v}_{Rw} = \mathbf{v} - l_R \mathbf{w} \times \mathbf{a} \tag{4.23}$$

Relating (3.1) and (4.23), we can obtain

$$\mathbf{v}_{Rw} = \mathbf{v}_{Lw} + l_{LR} \mathbf{w} \times \mathbf{a} \tag{4.24}$$

where $l_{LR} = L_L - L_R$. Substituting (4.12) and (4.13) into (4.24), we have,

$$\begin{bmatrix} s_{R1}(d_4 s_{R23} - a_2 c_{R2}) & -c_{R1}(d_4 c_{R23} + a_2 s_{R2}) & -d_4 c_{R1} c_{R23} \\ c_{R1}(-d_4 s_{R23} + a_2 c_{R2}) & -s_{R1}(d_4 c_{R23} + a_2 s_{R2}) & -d_4 s_{R1} c_{R23} \\ 0 & -d_4 s_{R23} + a_2 c_{R2}) & -d_4 s_{R23} \end{bmatrix} \begin{bmatrix} \dot{\theta}_{R1} \\ \dot{\theta}_{R2} \\ \dot{\theta}_{R3} \end{bmatrix} =$$
$$\begin{bmatrix} s_{L1}(d_4 s_{L23} - a_2 c_{L2}) & -c_{L1}(d_4 c_{L23} + a_2 s_{L2}) & -d_4 c_{L1} c_{L23} \\ -c_{L1}(d_4 s_{L23} - a_2 c_{L2}) & -s_{L1}(d_4 c_{L23} + a_2 s_{L2}) & -d_4 s_{L1} c_{L23} \\ 0 & -d_4 s_{L23} + a_2 c_{L2}) & -d_4 s_{L23} \end{bmatrix} \begin{bmatrix} \dot{\theta}_{L1} \\ \dot{\theta}_{L2} \\ \dot{\theta}_{L3} \end{bmatrix} + l_{LR} \begin{bmatrix} \omega_n \\ \omega_o \\ 0 \end{bmatrix} \times \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{4.25}$$

This vector equation can be rewritten in three scalar equations with three unknowns $\dot{\theta}_{R2}$, ω_n, and ω_o. From the x component of (4.25), we get:

$$\begin{aligned} w_o = \frac{1}{l_{LR} a_z} [& s_{R1}(d_4 s_{R23} - a_2 c_{R2})\dot{\theta}_{R1} - c_{R1}(d_4 c_{R23} + a_2 s_{R2})\dot{\theta}_{R2} - d_4 c_{R1} c_{R23}\dot{\theta}_{R3} \\ & - s_{L1}(d_4 s_{L23} - a_2 c_{L2})\dot{\theta}_{L1} + c_{L1}(d_4 c_{L23} + a_2 s_{L2})\dot{\theta}_{L2} + d_4 c_{L1} c_{L23}\dot{\theta}_{L3}] \end{aligned} \tag{4.26}$$

From the y component of (4.25), we get:

$$\begin{aligned} w_n = \frac{1}{l_{LR} a_z} [& -c_{R1}(d_4 s_{R23} - a_2 c_{R2})\dot{\theta}_{R1} + s_{R1}(d_4 c_{R23} + a_2 s_{R2})\dot{\theta}_{R2} + d_4 s_{R1} c_{R23}\dot{\theta}_{R3} \\ & + c_{L1}(d_4 s_{L23} - a_2 c_{L2})\dot{\theta}_{L1} - s_{L1}(d_4 c_{L23} + a_2 s_{L2})\dot{\theta}_{L2} - d_4 s_{L1} c_{L23}\dot{\theta}_{L3}] \end{aligned} \tag{4.27}$$

From the z component of (4.25), we get the passive joint rate:

$$\dot{\theta}_{R2} = \frac{1}{-d_4 s_{R23} + a_2 c_{R2}} [d_4 s_{R23}\dot{\theta}_{R3} + (-d_4 s_{L23} + a_2 c_{L2})\dot{\theta}_{L2} - d_4 s_{L23}\dot{\theta}_{L3} + l_{LR} a_y \omega_n - l_{LR} a_x \omega_o] \tag{4.28}$$

Substituting (4.26) and (4.27) into (4.28), $\dot{\theta}_{R2}$ can then be expressed in term of active joint rates as:

$$\begin{aligned} \dot{\theta}_{R2} &= e_1\dot{\theta}_{L1} + e_2\dot{\theta}_{L2} + e_3\dot{\theta}_{L3} + e_4\dot{\theta}_{R1} + e_5\dot{\theta}_{R3} \\ &= \mathbf{e}^{\mathrm{T}}\dot{\mathbf{q}} \end{aligned} \tag{4.29}$$

where $e_1 = \frac{1}{a_z(-d_4 s_{R2} + a_2 c_{R2})}(a_y c_{L1} - a_x s_{L1})(d_4 s_{L23} - a_2 c_{L2})$

$$e_2 = \frac{1}{a_z(-d_4 s_{R2} + a_2 c_{R2})}[a_z(d_4 s_{L23} - a_2 c_{L2}) - (a_y s_{L1} + a_x c_{L1})(d_4 c_{L23} - a_2 s_{L2})]$$

$$e_3 = \frac{1}{a_z(-d_4 s_{R2} + a_2 c_{R2})}[a_z d_4 s_{L23} - d_4 c_{L23}(a_y s_{L1} - a_x c_{L1})]$$

$$e_4 = \frac{1}{a_z}(a_y c_{R1} + a_x s_{R1})$$

$$e_5 = \frac{1}{a_z(-d_4 s_{R2} + a_2 c_{R2})}[a_z d_4 s_{R23} - d_4 c_{R23}(a_y s_{R1} - a_x c_{R1})]$$

Substituting (4.29) into (4.26) and (4.27), we can obtain the angular velocity **w**. Substituting **w** into (4.11), we can get the hand velocity **v**.

The forward velocity analysis of the platform robot is to find the platform velocity given the joint rates. This can be accomplished by relating the velocity of the three wrists. Similar to (4.23) we can relate the velocity of two wrists from (4.11) as:

$$\mathbf{v}_{iw} = \mathbf{v}_{jw} + \mathbf{w} \times (\mathbf{l}_i - \mathbf{l}_j) \tag{4.30}$$

where *i* and *j* indicate wrists on different arms.

Relating $\mathbf{v}_{1w}$ and $\mathbf{v}_{3w}$, $\mathbf{v}_{1w}$ and $\mathbf{v}_{2w}$ in (4.30) respectively, these two vector equations can be rewritten in six scalar equations with six unknowns $\dot{\theta}_{12}$, $\dot{\theta}_{22}$, $\dot{\theta}_{32}$, ω_n, ω_o, and ω_a. Notice that relating $\mathbf{v}_{2w}$ and $\mathbf{v}_{3w}$ in (4.30) we can get a third but redundant equation. Once the angular velocity is solved, the platform velocity can be obtain from (4.11).

4.4 Case Studies

MATLAB programs were written to verify rate control algorithm of linkage and platform robots. The motion simulation of a platform robot is performed in the first case. The robot's upper and lower link are 1.0 m long in each arm. The feet on the platform and base are each on an equilateral triangle. Each side of the base and platform triangles are 4.0 m and 2.0 m respectively.

For inverse velocity, the platform velocity is given as $\begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} = [0,0,-1;0,0,0]^T$. The platform center is located at [0,0,1.2], with yaw, pitch, and roll angles as [0,0,10]. The joint rate calculated by equation (4.10) is $\dot{\mathbf{q}} = [0,1.2627;0,1.2627;0,1.2627]$, and that by equation (4.14) is $\dot{\mathbf{q}}$=[0,-1.0460,1.2627;0,1.0460,1.2627;0,-1.0460,1.2627]. Active joint rates from these two results agree. The joint rates can also be verified by physical interpretation. Since each wrist has a downward motion only, the base is not rotating and $\dot{\theta}_{i1}$ should be zero as calculated.

In the second case, a linkage robot is studied. The robot's upper and lower links are 1.0 m long in each arm, and the distance from the tool to the left wrist is 1.3 m. Distance between the two bases and between the two wrists are both 0.6 m. For inverse velocity, the tool velocity is given as [0, 1, 0;' 0, 0]. The tool position is at [1.1, 0, 1.3],

with yaw, pitch, roll angles as [0, 85, 0]. The rate calculated by equation (4.22) is $\dot{\mathbf{q}} = [5.1268, 0, 0; 5.1268, 0]$, and the rate calculated by (4.14) is $\dot{\mathbf{q}} = [5.1268, 0, 0; 5.1268, 0, 0]$. Eliminating the passive joint rate from the latter method, the two methods yield identical results. The results can also be verified by physical interpretation. Since the tool is moving in the transverse direction only, only $\dot{\theta}_{i1}$ is non zero.

4.5 Discussion

As stated in section 4.3, five actuators can be used for the 5 DOF linkage robot without becoming a singular configuration. For better load distribution, the right shoulder can also become active, and the robot becomes redundant actuation.

Since each column of **S** can be expressed by (4.10), the **S** matrix can easily be expanded to a rank of 6x5 with six active joints. $\mathbf{S}^{-1}$ will then be replaced by the pseudoinverse. Because of the pseudoinverse, the redundancy will yield a minimum norm of the wrist force, and in turn a minimum norm of the joint torque.

The resolved force control presented in this chapter can be extended to any serial-parallel manipulator. For example, a planar version of three articulated-arm platform robot [Kokkinis and Stoughton 1988] and a planar version of the linkage robot [Wang and You 1992] can be easily adapted derived using wrenches. Three DOF spatial platform manipulators [Yang 1995] can also be derived this way.

CHAPTER FIVE

THE DYNAMIC MODEL OF A LINKAGE ROBOT

The Lagrange formulation is used to derive the dynamic models of a linkage robot from their potential and kinematics energy. The dynamic model of the linkage robot will be developed by hypothetically cutting the linkage at the two wrists into three segments and finding joint torques affected by each of the three parts.

5.1 The Dynamic Model of a Serial Robot

The equation of motion of a serial robot can be expressed as:

$$\tau_i = \sum_{j=1}^{n} D_{ij}(q)\ddot{q}_j + \sum_{k=1}^{n}\sum_{j=1}^{n} C_{kj}^{i}(q)\dot{q}_k\dot{q}_j + h_i(q) + b_i(q) \tag{5.1}$$

where τ_i is the joint torque,

$\dot{q}$ the joint rate,

$\ddot{q}$ the joint acceleration,

D_{ij} the coupled inertia between Joints i and j,

C_{kj}^{i} the Coriolis and centrifugal terms at Joint i,

h_i the gravitational torque at Joint i,

b_i the frictional torque at Joint i.

D_{ij} of a multiple link system can be derived from the kinetic energy; h_i can be derived from the potential energy and the Lagrangian equation; C_{kj}^{i} can be derived from D_{ij} using the Lagrangian method.

The kinetic energy T of a serial robot with n links can be expressed as:

$$T(q,\dot{q}) = \frac{1}{2}\sum_{k=1}^{n} \mathbf{w}_k^{\mathrm{T}}\mathbf{I}_k\mathbf{w}_k + \frac{1}{2}\sum_{k=1}^{n} \mathbf{v}_k^{\mathrm{T}} m_k \mathbf{v}_k \tag{5.2}$$

where $\mathbf{I}_k$ the inertial tensor of Link k,

m_k the mass of Link k,

$\mathbf{w}_k$ the angular velocity of Link k,

$\mathbf{v}_k$ the centroidal velocity of Link k.

$\mathbf{w}_k$ and $\mathbf{v}_k$ are functions of joint rates and angles, and can be expressed as:

$$\mathbf{w}_k = \mathbf{A}^k\dot{\mathbf{q}} \tag{5.3}$$

$$\mathbf{v}_k = \mathbf{B}^k\dot{\mathbf{q}}$$

where $\mathbf{A^K}$ and $\mathbf{B^k}$ are the upper and lower parts of the manipulator Jacobian and are associated with translation and rotation respectively. Substituting (5.3) into (5.2), we can get

$$T(q,\dot{q}) = \frac{1}{2}\dot{\mathbf{q}}^{\mathbf{T}}\sum_{k=1}^{n}[(\mathbf{A^k})^{\mathbf{T}} m_k \mathbf{A^k} + (\mathbf{B^k})^{\mathbf{T}}\mathbf{D}_{\mathbf{k}}\mathbf{B^k}]\dot{\mathbf{q}} = \frac{1}{2}\dot{\mathbf{q}}^{\mathbf{T}}\mathbf{D}\dot{\mathbf{q}} \tag{5.4}$$

where $\mathbf{D}_{\mathbf{k}}$ is the *k*th link's inertial tensor, and **D** is the manipulator inertial tensor.

The potential energy U can be expressed as:

$$U(q) = \sum_{k=1}^{n} m_k g^T c^k \tag{5.5}$$

where m_k is the mass of link k, and c^k is the position of the center of mass of link k in the base coordinates.

The Lagrangian L is defined as the difference between the kinetic and potential energy of a mechanical system.

$$L(q,\dot{q}) = T(q,\dot{q}) - U(q) \tag{5.6}$$

The equation of motion of a serial robotic arm can be formulated in terms of the Lagrangian function as follows:

$$\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \left(\frac{\partial L}{\partial q_i}\right) \tag{5.7}$$

Substituting (5.4), (5.5), and (5.6) into (5.7), we get:

$$\tau_i = \sum_{j=1}^{n} D_{ij}\ddot{q}_j + \sum_{k=1}^{n}\sum_{j=1}^{n} C_{kj}^{i}\dot{q}_k\dot{q}_j + h_i \tag{5.8}$$

Gravitational term h_i can expressed as:

$$h_i = -g_0\sum_{j=i}^{n} m_j \bar{c}^j \tag{5.9}$$

where g_0 is the gravitational constant, and $\bar{c}^j$ is the position of the center of mass of the *jth* link.

The Coriolis and centrifugal terms C_{kj}^{i} can be related to D_{ij} as:

$$C_{kj}^{i} = \frac{\partial D_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial D_{kj}}{\partial q_i} \tag{5.10}$$

5.2 The Dynamic Model of a Linkage Robot

The dynamic model of a linkage robot can be analyzed by hypothetically cutting the linkage at the two wrists into three segments. The joint torque can then be obtained as:

$$\tau_i = \tau_i^L + \tau_i^c + \tau_i^R + b_i \tag{5.11}$$

where b_i is the joint friction; τ_i^L, τ_i^c, and τ_i^R are joint torques affected by the left arm, coupler arm, , right arm respectively.

The left arm can be analyzed like a serial robot. The right arm is also a serial arm, but with a passive shoulder joint. Therefore dynamic terms associated with the passive joint rate need to be related to active joint rates. Similarly, to find dynamic terms associated with the coupler link, the coupler velocity should be expressed in terms of active joint rates.

5.3 Dynamic Terms Contributed by the Left Arm

The Jacobian matrix of left arm's first three joints can be expressed as:

$$\mathbf{J^{L4}} = \begin{bmatrix} -a_2 s_{L1} c_{L2} + r_4 s_{L1} s_{L23} & -a_2 c_{L1} s_{L2} - r_4 c L_1 c_{L23} & -r_4 c_{L1} c_{L23} \\ a_2 c_{L1} c_{L2} - r_4 c_{L1} s_{L23} & -a_2 s_{L1} s_{L2} - r_4 c_{L1} s_{L23} & -r_4 s_{L1} c_{L23} \\ 0 & a_2 c_{L2} - r_4 s_{L23} & -r_4 s_{L23} \\ 0 & s_{L1} & s_{L1} \\ 0 & -c_{L1} & -c_{L1} \\ 1 & 0 & 0 \end{bmatrix} \tag{5.12}$$

The inertia tensor of the left arm can be expressed as:

$$\mathbf{D^L} = \begin{bmatrix} D_{11}^L & 0 & 0 \\ 0 & D_{22}^L & D_{23}^L \\ 0 & D_{23}^L & D_{33}^L \end{bmatrix} \tag{5.13}$$

where $D_{11}^L = r_2^2 m_2 c_{L2}^2 + I_2 c_{L2}^2 + m_4 (r_2 s_{L23} - a_2 c_{L2})^2 + I_4 s_{L23}^2$

$D_{22}^L = r_2^2 m_2 + I_2 + m_4 (r_4^2 - 2 r_4 a_2 s_{L3} + a_2^2) + I_4$

$D_{23}^L = m_4 (r_4^2 - r_4 a_2 s_{L3}) + I_4$

$D_{33}^L = r_4^2 + I_4$

Once D_{ij} is calculated, the corresponding C_{kj}^L of the left arm can be derived by (5.10). The gravitational terms are:

$$h_1^L = 0 \tag{5.14}$$

$$h_2^L = g_0 [m_2 r_2 c_{L2} + m_4 (a_2 c_{L2} - r_4 s_{L23})] \tag{5.15}$$

$$h_3^L = -g_0 m_4 r_4 s_{L23} \tag{5.16}$$

Substituting (5.13), (5.14), (5.15), and (5.16) into (5.8), the joint torques contributed by the left arm are:

$$\begin{aligned}
\tau_1^L &= D_{11}^L \ddot{q}_1 + C_{12}^{L1} \dot{q}_1 \dot{q}_2 + C_{13}^{L1} \dot{q}_1 \dot{q}_3 \\
\tau_2^L &= D_{22}^L \ddot{q}_2 + D_{23}^L \ddot{q}_3 + C_{11}^{L2} \dot{q}_1^2 + C_{32}^{L2} \dot{q}_2 \dot{q}_3 + C_{33}^{L2} \dot{q}_3^2 + h_2^L \\
\tau_3^L &= D_{32}^L \ddot{q}_2 + D_{33}^L \ddot{q}_3 + C_{11}^{L3} \dot{q}_1^2 + C_{22}^{L3} \dot{q}_3^2 + 2C_{32}^{L3} \dot{q}_2 \dot{q}_3 + h_3^L
\end{aligned} \tag{5.17}$$

where
$$C_{12}^{L1} = -2(r_2^2 m_2 + I_2) c_{L2} s_{L2} + 2m_4 (r_4 s_{L23} - a_2 c_{L2})(r_4 c_{L23} + a_2 s_{L2}) + 2I_4 s_{L23} c_{L23}$$
$$C_{13}^{L1} = 2m_4 r_4 (r_4 s_{L23} - a_2 c_{L2}) c_{L23} + 2I_4 s_{L23} c_{L23}$$
$$C_{11}^{L2} = (r_2^2 m_2 + I_2) c_{L2} s_{L2} + m_4 (r_4 s_{L23} - a_2 c_{L2})(r_4 c_{L23} + a_2 s_{L2}) + I_4 s_{L23} c_{L23}$$
$$C_{32}^{L2} = -2m_4 r_4 a_2 c_3$$
$$C_{33}^{L2} = -m_4 r_4 a_2 c_3$$
$$C_{11}^{L3} = 2m_4 (r_4 s_{L23} - a_2 c_{L2}) r_4 c_{L23} + 2I_4 s_{L23} c_{L23}$$
$$C_{22}^{L3} = -m_4 r_4 a_2 c_{L3}$$
$$C_{32}^{L3} = \frac{1}{2} m_4 r_4 a_2 c_{L3}$$

5.4 Dynamic Terms Contributed by the Coupler Link

The coupler arm's velocity $\mathbf{w}_c$ and $\mathbf{v}_c$ can be expressed in terms of active joint rate as (3.20):

$$\begin{bmatrix} \mathbf{v}_c \\ \mathbf{w}_c \end{bmatrix} = (\mathbf{S}_5^{\mathrm{T}})^{-1} \dot{\mathbf{q}}_5 \tag{5.18}$$

where $\mathbf{S}_5$ is defined in (3.19). From (5.18), $\mathbf{w}_C$ and $\mathbf{v}_C$ can be expressed as:

$$\mathbf{v}_C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} (\mathbf{S}_5^{\mathrm{T}})^{-1} \dot{\mathbf{q}}_5 \tag{5.19}$$

$$\mathbf{w}_C = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} (\mathbf{S}_5^{\mathrm{T}})^{-1} \dot{\mathbf{q}}_5 \tag{5.20}$$

Then the inertia tensor can be derived from the kinetic energy as:.

$$T_C = \frac{1}{2} \mathbf{v}_C^{\mathrm{T}} m_C \mathbf{v}_C + \frac{1}{2} \mathbf{w}_C^{\mathrm{T}} \mathbf{I}_C \mathbf{w}_C = \frac{1}{2} \dot{\mathbf{q}}_5^{\mathrm{T}} \mathbf{D}_C \dot{\mathbf{q}}_5 \tag{5.21}$$

Substituting (5.19) and (5.20) into (5.21), we have:

$$\mathbf{D}_C = (\mathbf{S}_5)^{-1}\begin{bmatrix} m_c & 0 & 0 & 0 & 0 \\ 0 & m_c & 0 & 0 & 0 \\ 0 & 0 & m_c & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}(\mathbf{S}_5^T)^{-1} + (\mathbf{S}_5)^{-1}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_c & 0 \\ 0 & 0 & 0 & 0 & I_c \end{bmatrix}(\mathbf{S}_5^T)^{-1}$$

$$= (\mathbf{S}_5)^{-1}\mathbf{B}(\mathbf{S}_5^T)^{-1}$$

$$\text{where } \mathbf{B} = \begin{bmatrix} m_c & 0 & 0 & 0 & 0 \\ 0 & m_c & 0 & 0 & 0 \\ 0 & 0 & m_c & 0 & 0 \\ 0 & 0 & 0 & I_c & 0 \\ 0 & 0 & 0 & 0 & I_c \end{bmatrix} \tag{5.22}$$

The velocity coupling term $\frac{\partial \mathbf{D}_C}{\partial q_i}$ in (5.10) can be obtained by applying the product rule to (5.22) as:

$$\frac{\partial \mathbf{D}_C}{\partial q_i} = \frac{\partial (\mathbf{S}_5)^{-1}}{\partial q_i}\mathbf{B}(\mathbf{S}_5^T)^{-1} + (\mathbf{S}_5)^{-1}\mathbf{B}\frac{\partial (\mathbf{S}_5^T)^{-1}}{\partial q_i} \tag{5.23}$$

To get $\frac{\partial \mathbf{S}_5^{-1}}{\partial q_i}$ and $\frac{\partial (\mathbf{S}_5^T)^{-1}}{\partial q_i}$ in this equation, we can use the identity matrix $\mathbf{I}_5 = \mathbf{S}_5\mathbf{S}_5^{-1}$. Taking derivative of $\mathbf{I}_5$ with respect to q_i, we have:

$$\frac{\partial \mathbf{I}_5}{\partial q_i} = \frac{\partial \mathbf{S}_5}{\partial q_i}\mathbf{S}_5^{-1} + \mathbf{S}_5\frac{\partial \mathbf{S}_5^{-1}}{\partial q_i} = 0.$$

$$\text{Therefore, } \frac{\partial \mathbf{S}_5^{-1}}{\partial q_i} = -\mathbf{S}_5^{-1}\frac{\partial \mathbf{S}_5}{\partial q_i}\mathbf{S}_5^{-1} \tag{5.24}$$

Similarly, using $\mathbf{I}_5 = (\mathbf{S}_5^T)(\mathbf{S}_5^T)^{-1}$, we can derive

$$\frac{\partial (\mathbf{S}_5^T)^{-1}}{\partial q_i} = -(\mathbf{S}_5^T)^{-1}\frac{\partial \mathbf{S}_5^T}{\partial q_i}(\mathbf{S}_5^T)^{-1}. \tag{5.25}$$

Since matrix inversion is computationally intensive, we would like to relate $(\mathbf{S}_5^T)^{-1}$ in (5.25) to $(\mathbf{S}_5^{-1})^T$. We can use the identity matrix $\mathbf{I}_5 = (\mathbf{S}_5^T)^{-1}\mathbf{S}_5^T$, (5.26)

and $\mathbf{I}_5 = (\mathbf{S}_5\mathbf{S}_5^{-1})^T$ (5.27)

Applying the transpose identity to (5.27), we get $\mathbf{I}_5 = (\mathbf{S}_5^{-1})^{\mathrm{T}}\mathbf{S}_5^{\mathrm{T}}$, (5.28)

Comparing (5.26) and (5.28), we get $(\mathbf{S}_5^{\mathrm{T}})^{-1} = (\mathbf{S}_5^{-1})^{\mathrm{T}}$ (5.29)

Substituting (5.29) into (5.25), we have

$$\frac{\partial(\mathbf{S}_5^T)^{-1}}{\partial q_i} = -(\mathbf{S}_5^{-1})^T \frac{\partial \mathbf{S}_5^T}{\partial q_i}(\mathbf{S}_5^{-1})^T = -\left(\mathbf{S}_5^{-1}\frac{\partial \mathbf{S}_5}{\partial q_i}\mathbf{S}_5^{-1}\right)^T \tag{5.30}$$

Substituting (5.24) into (5.30), we have

$$\frac{\partial(\mathbf{S}_5^T)^{-1}}{\partial q_i} = \left(\frac{\partial \mathbf{S}_5^{-1}}{\partial q_i}\right)^T \tag{5.31}$$

Equation (5.23) can then be written as:

$$\frac{\partial \mathbf{D}_C}{\partial q_i} = \frac{\partial \mathbf{S}_5^{-1}}{\partial q_i}\mathbf{B}(\mathbf{S}_5^{-1})^T + \mathbf{S}_5^{-1}\mathbf{B}\left[\frac{\partial(\mathbf{S}_5^{-1})}{\partial q_i}\right]^T \tag{5.32}$$

Substituting (5.32) into (5.10), C_{kj}^i can then be obtained.

To get the gravitational term, the coupler's centroid coordinates can be derived as:

$$\mathbf{p}_{\mathbf{C}} = \mathbf{p}_{\mathbf{Lw}} + \frac{l_{CL}}{k}(\mathbf{p}_{\mathbf{Rw}} - \mathbf{p}_{\mathbf{Lw}}) \tag{5.33}$$

where k is the distance between the two wrists, l_{CL} is the distance from the mass center of coupler arm to the left wrist, and $\mathbf{p}_{\mathbf{Lw}}$ and $\mathbf{p}_{\mathbf{Rw}}$ are the coordinates of the left and right wrists respectively.

The z component of the $\mathbf{p}_{\mathbf{C}}$ can be expressed as:

$$p_{Cz} = \frac{1}{k}[(l_{CL}+k)(a_2 s_{L2} + d_4 c_{L23} - l_{CL}(l_{R2}s_{R2} + l_{R3}s_{R23})] \tag{5.34}$$

Therefore the gravitational terms can be derived as:

$$h_i^C = g_0 m_C \frac{\partial p_{Cz}}{\partial q_i} \tag{5.35}$$

Then the joint torque contributed by the coupler arm is:

$$\tau_i^C = \sum_{j=1}^{n} D_{ij}^C(q)\ddot{q}_j + \sum_{k=1}^{n}\sum_{j=1}^{n} C_{kj}^{Ci}(q)\dot{q}_k\dot{q}_j + h_i^C(q) \tag{5.36}$$

5.5 Dynamic Terms Contributed by the right arm

The right arm of the linkage robot has a similar structure as the left arm. So the kinetic energy for the right arm can be written as:

$$T_R = \frac{\dot{\mathbf{q}}_\mathbf{R}^\mathbf{T}\mathbf{D}\,\dot{\mathbf{q}}_\mathbf{R}}{\mathbf{2}} = D_{11}\dot{q}_{R1}^2 + D_{22}\dot{q}_{R2}^2 + D_{33}\dot{q}_{R3}^2 + 2D_{23}\dot{q}_{R2}\dot{q}_{R3} \tag{5.37}$$

From (4.29) $\dot{q}_{R2}$ can be expressed in term of active joint rates as:

$$\dot{q}_{R2} = \mathbf{e}^\mathbf{T}\dot{\mathbf{q}} = \dot{\mathbf{q}}^\mathbf{T}\mathbf{e} \tag{5.38}$$

T_R can then be written as:

$$T_R = D_{11}\dot{q}_{R1}^2 + D_{22}\dot{\mathbf{q}}^\mathbf{T}\mathbf{e}\mathbf{e}^\mathbf{T}\dot{\mathbf{q}} + D_{33}\dot{q}_{R3}^2 + 2D_{23}\mathbf{e}^\mathbf{T}\dot{\mathbf{q}}\dot{q}_{R3} \tag{5.39}$$

$\dot{q}_{R3}$ can be expressed in vector form and then (5.39) can be rewritten as:

$$T_R = D_{11}\dot{q}_{R1}^2 + D_{22}\dot{\mathbf{q}}^\mathbf{T}\mathbf{e}\mathbf{e}^\mathbf{T}\dot{\mathbf{q}} + D_{33}\dot{q}_{R3}^2 + 2D_{23}\dot{\mathbf{q}}^\mathbf{T}\mathbf{e}\begin{bmatrix}0 & 0 & 0 & 0 & 1\end{bmatrix}\dot{\mathbf{q}} \tag{5.40}$$

Collecting terms, the kinetic energy can be expressed as:

$$T_R = \dot{\mathbf{q}}^\mathbf{T}\mathbf{D}^\mathbf{R}\dot{\mathbf{q}} \tag{5.41}$$

where $$\mathbf{D}^\mathbf{R} = \begin{bmatrix} D_{11} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D_{33} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2D_{23}e_5 \end{bmatrix} + D_{22}\mathbf{e}\mathbf{e}^\mathbf{T} \tag{5.42}$$

With inertia term $\mathbf{D}^\mathbf{R}$, C_{kj}^i can then be derived using (5.10).

Joint torques contributed by the right arm can be expressed as:

$$\tau_i^R = \sum_{j=1}^{n} D_{ij}^R(q)\ddot{q}_j + \sum_{k=1}^{n}\sum_{j=1}^{n} C_{kj}^{Ri}(q)\dot{q}_k\dot{q}_j + h_i^R(q) \tag{5.43}$$

Substituting (5.17), (5.36), and (5.43) into (5.11), we can get the joint torque τ.

5.6 Discussion

The contribution of this chapter is to develop a dynamic model that all dynamic terms can be explicitly shown. With these terms we can evaluate the numerical value of

each term with a given robot geometry and under a specific path. If certain terms like C_{ij} are insignificant, we can neglect them in the torque calculation.

The challenge of the linkage robot dynamic model developed in this chapter is the calculation of (5.23) in the coupler arm and the calculation of (5.42) in the right arm. Each column of $\mathbf{S_5}$ in (5.23) is a sinusoidal function of joint angles, and the derivatives of $\mathbf{S_5}$ will be complex. The elements of vector **e** in (5.37) are also sinusoidal functions as shown in (4.29), and therefore the computation is time consuming.

CHAPTER SIX

COLLISION DETECTION OF A LINKAGE ROBOT

An efficient collision detection algorithm is developed for linkage and platform robots. To detect the collision of multiple arms of a serial-parallel manipulator, the collision of any two links of the articulated arms need to be investigated first. Cylinders are used to model links, and line equations with parameters are used to represent center-lines of these cylinders. If the distance between two center-lines is less than the sum of radii of these two cylinders, and the feet of the common normal of the two center-lines are inside the links, collision happens. A MATLAB program was written to visualize and verify the collision detecting algorithm.

6.1 Collision between Two Links

Two links are modeled as cylinders and are showed in Figure 6.1 a. Figure 6.1 b shows two line segments representing the cylinders' center-lines. $\mathbf{P}_{11}$, $\mathbf{P}_{12}$, $\mathbf{P}_{21}$, and $\mathbf{P}_{22}$ are respective endpoints of the line segments L_1 and L_2. The relationship between the two lines containing L_1 and L_2 can be skew or coplanar, and the coplanar lines can be parallel, intersecting, and collinear.

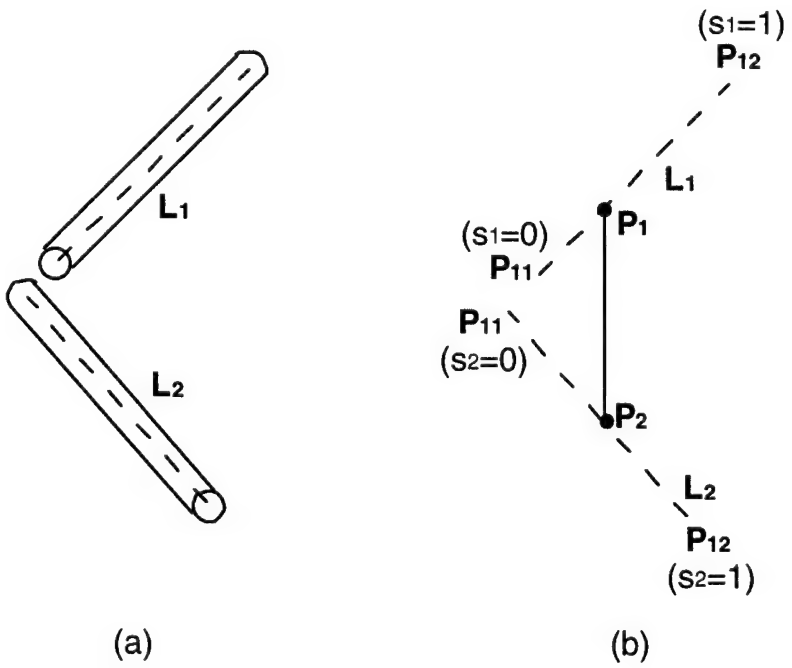

Figure 6. 1 Two links and their common normal

If the two lines containing line segments L_1 and L_2 are skew, there is a unique common normal L to these two lines. Osgood and Graustein [1960] used direction components of the two lines to decide their common normal, and calculated the distance between the two lines with the coordinates of one point on each line. In a robot's motion control, the coordinates of endpoints of each link (line segment) can always be calculated from joint angles. Therefore coordinates of $\mathbf{P}_{11}$, $\mathbf{P}_{12}$, $\mathbf{P}_{21}$, and $\mathbf{P}_{22}$, instead of direction components, will be used in the algorithm derivation.

Since we are concerned about the collision of links (line segments), not just two lines, the location of the feet of the common normal, in addition to the distance between them, is important. To decide if a common normal's foot, $\mathbf{P}_1$ or $\mathbf{P}_2$, is within a line segment, they can be represented by a parametric form as:

$$\mathbf{P}_1 = \mathbf{P}_{11} + s_1(\mathbf{P}_{12} - \mathbf{P}_{11}) \tag{6.1}$$

$$\mathbf{P}_2 = \mathbf{P}_{21} + s_2(\mathbf{P}_{22} - \mathbf{P}_{21}) \tag{6.2}$$

where s_1 and s_2 are the parameters. When $0 \le s_i \le 1$, a normal foot is within the line segment of L_i. The normal condition between the common normal L ($\mathbf{P}_1$-$\mathbf{P}_2$) and the two line segments L_1 ($\mathbf{P}_{11}$-$\mathbf{P}_{12}$) and L_2 ($\mathbf{P}_{21}$-$\mathbf{P}_{22}$) can be expressed by inner products as:

$$(\mathbf{P}_2 - \mathbf{P}_1) \bullet (\mathbf{P}_{12} - \mathbf{P}_{11}) = 0 \tag{6.3}$$

$$(\mathbf{P}_2 - \mathbf{P}_1) \bullet (\mathbf{P}_{22} - \mathbf{P}_{21}) = 0 \tag{6.4}$$

Substituting (6.1) and (6.2) into (6.3) and (6.4) respectively, we get:

$$((\mathbf{P}_{21} - \mathbf{P}_{11}) + s_2(\mathbf{P}_{22} - \mathbf{P}_{21}) - s_1(\mathbf{P}_{12} - \mathbf{P}_{11})) \bullet (\mathbf{P}_{12} - \mathbf{P}_{11}) = 0 \tag{6.5}$$

$$((\mathbf{P}_{21} - \mathbf{P}_{11}) + s_2(\mathbf{P}_{22} - \mathbf{P}_{21}) - s_1(\mathbf{P}_{12} - \mathbf{P}_{11})) \bullet (\mathbf{P}_{22} - \mathbf{P}_{21}) = 0 \tag{6.6}$$

These two equations can be expressed in terms of parameters s_1 and s_2 as:

$$-a_1 s_1 + b s_2 = c_1 \tag{6.7}$$

$$-b s_1 + a_2 s_2 = c_2 \tag{6.8}$$

where $a_1 = (\mathbf{P}_{12} - \mathbf{P}_{11}) \bullet (\mathbf{P}_{12} - \mathbf{P}_{11})$

$a_2 = (\mathbf{P}_{22} - \mathbf{P}_{21}) \bullet (\mathbf{P}_{22} - \mathbf{P}_{21})$

$b = (\mathbf{P}_{22} - \mathbf{P}_{21}) \bullet (\mathbf{P}_{12} - \mathbf{P}_{11})$

$c_1 = -(\mathbf{P}_{21} - \mathbf{P}_{11}) \bullet (\mathbf{P}_{12} - \mathbf{P}_{11})$

$c_2 = (\mathbf{P}_{21} - \mathbf{P}_{11}) \bullet (\mathbf{P}_{12} - \mathbf{P}_{11})$

When (6.7) and (6.8) are linearly independent, we can solve for s_1 and s_2 as:

$$s_1 = (c_1 a_2 - c_2 b) / D \tag{6.9}$$

$$s_2 = (-c_2 a_1 + c_1 b) / D \tag{6.10}$$

where $D = -a_1 a_2 + b^2$

If both s_1 and s_2 are within limits, the two cylinders collide when the distance between the two feet is less than the sum of link diameters:

$$|P_2 - P_1| \le 0.5(d_2 + d_1) \tag{6.11}$$

where d_2 and d_1 are respective diameters of the two cylinders.

The link collision may happen when a foot of the common normal is outside a link, and the end face of one link collides the other link. Figure 6.2 shows the end-view of link 2, with the center-line of link 1 in a plane which is normal to link 2. As shown in Figure 6.2b, the foot of the common normal is outside the cylinder with a distance of $0.5d_2 \sin\theta$, and this distance is largest when $\theta = 90^\circ$, as shown in Figure 6.2c. To account for this type of collision, the range criterion is modified to:

$$\alpha_j \le s_j \le 1 + \alpha_j \qquad \text{j=1,2} \tag{6.12}$$

where α_j is the ratio of the radius of one link to the length of the other link: $\alpha_1 = d_2 / 2l_1$, and $\alpha_2 = d_1 / 2l_2$.

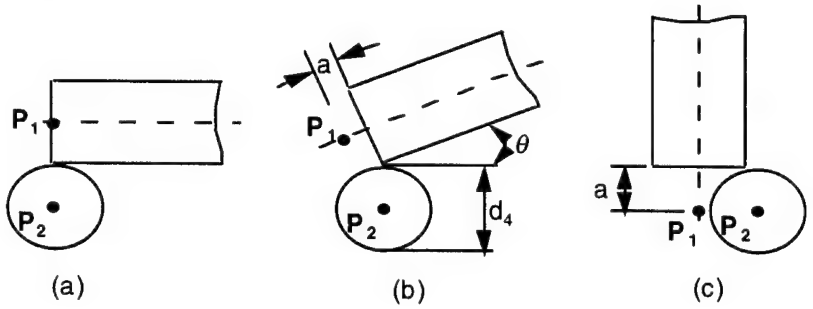

Figure 6. 2 Cases when a foot of the common normal is located outside the link

Equation (6.3) and (6.4) are not linearly independent if one of the following two conditions is true.

$$\mathbf{P}_2 - \mathbf{P}_1 = 0 \tag{6.13}$$

$$\mathbf{P}_{22} - \mathbf{P}_{21} = k(\mathbf{P}_{12} - \mathbf{P}_{11}) \tag{6.14}$$

If (6.13) is true, $\mathbf{P}_2 = \mathbf{P}_1$, and the common normal is degenerated to a point. This is when lines containing L_1 and L_2 are intersecting, and equations (6.3) and (6.4) become trivial. If (6.14) is true, the two line segments L_1 and L_2 are in the same direction, and this yields D=0 in (6.9) and (6.10). If the two lines are collinear, there is no common normal. If they are parallel, there are an infinite number of common normal lines.

Equation (6.13) or (6.14) is true when L_1 and L_2 are coplanar. The coplanar condition can be expressed as:

$$(\mathbf{P}_{12} - \mathbf{P}_{11}) \bullet [(\mathbf{P}_{22} - \mathbf{P}_{21}) \times (\mathbf{P}_{21} - \mathbf{P}_{11})] = \mathbf{0} \tag{6.15}$$

In this equation, $[(\mathbf{P}_{22} - \mathbf{P}_{21}) \times (\mathbf{P}_{21} - \mathbf{P}_{11})]$ is a line normal to L_2 and a line $(\mathbf{P}_{21} - \mathbf{P}_{11})$ connecting endpoints of L_1 and L_2. When this normal line is perpendicular to L_1, L_1 and L_2 are coplanar.

When lines containing L_1 and L_2 are coplanar and (6.14) is false, they are intersecting. The collision detection algorithm should be modified to detect the case when an end of a link hits another link. This type of collision, as shown in Figure 6.3 can be detected by monitoring the distance between a point $\mathbf{P}_2$ and a line segment L_1, as shown in Figure 6.4.

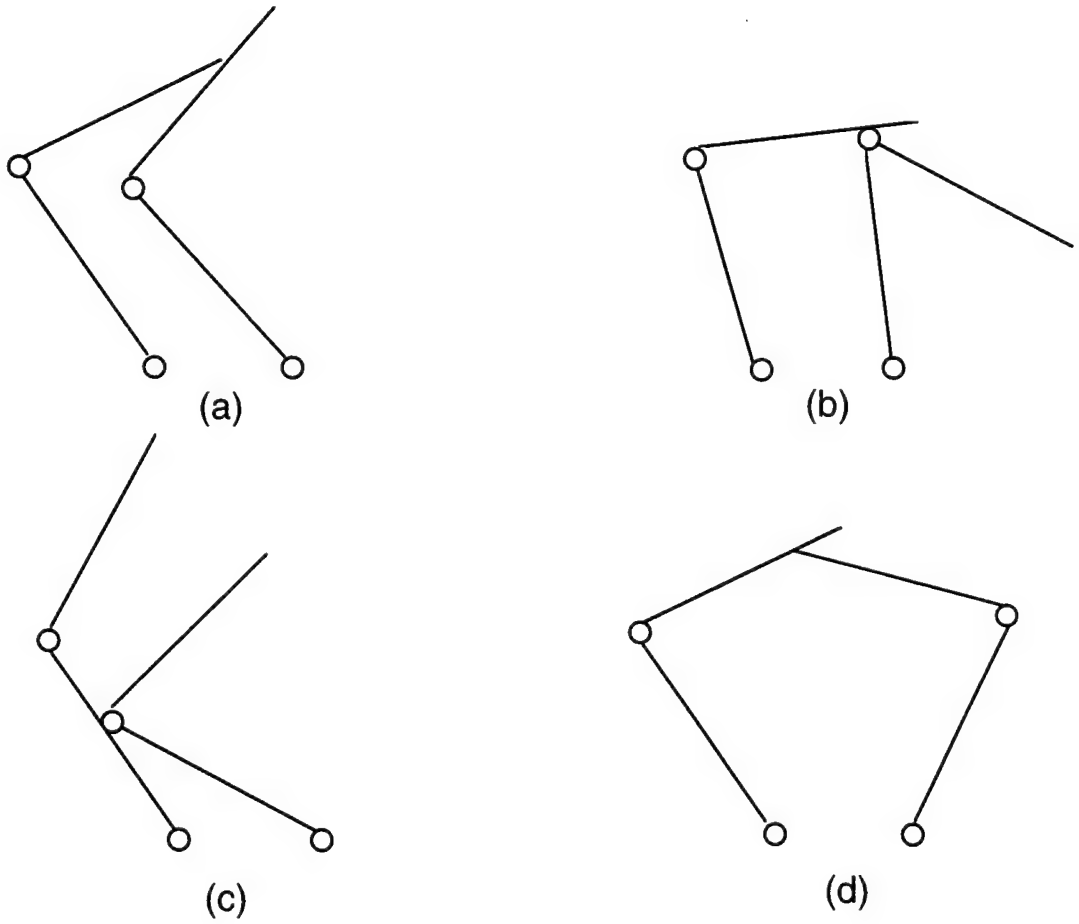

Figure 6. 3 Collisions of links in planar motion

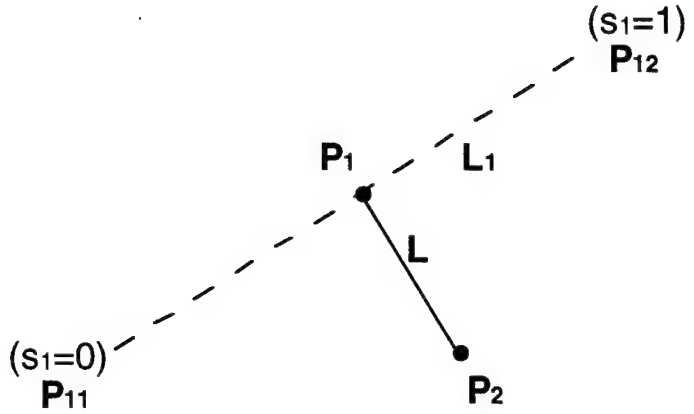

Figure 6. 4 The normal line between a point and a line

$\mathbf{P}_1$, the foot of the normal line from $\mathbf{P}_2$, can be expressed as a function of a parameter s_1, as in (6.1). Substituting (6.1) into (6.3), we get:

$$((\mathbf{P}_2 - \mathbf{P}_{11}) - s_1(\mathbf{P}_{12} - \mathbf{P}_{11})) \bullet (\mathbf{P}_{12} - \mathbf{P}_{11}) = 0 \qquad (6.16)$$

Point $\mathbf{P}_2$ here can be one of the endpoints of L_2, $\mathbf{P}_{21}$ or $\mathbf{P}_{22}$. Solving (6.16) for s_1, we get:

$$s_1 = ((\mathbf{P}_2 - \mathbf{P}_{11}) \bullet (\mathbf{P}_{12} - \mathbf{P}_{11})) / ((\mathbf{P}_{12} - \mathbf{P}_{11})) \bullet (\mathbf{P}_{12} - \mathbf{P}_{11})) \qquad (6.17)$$

When $0 \le s_1 \le 1$, the foot is within the line segment. The collision happens if the length of the normal line is short, as in (6.11). Similarly, to check if an endpoint of L_1 collides with L_2, we can substitute (6.2) into (6.4) to get:

$$s_2 = ((\mathbf{P}_1 - \mathbf{P}_{11}) \bullet (\mathbf{P}_{22} - \mathbf{P}_{21})) / ((\mathbf{P}_{22} - \mathbf{P}_{21})) \bullet (\mathbf{P}_{22} - \mathbf{P}_{21})) \qquad (6.18)$$

When $0 \le s_2 \le 1$, the foot is within the line segment. The collision happens if the length of the normal line is short, as in (6.11).

L_1 and L_2 are collinear when (6.14) is true and a line $(\mathbf{P}_{21} - \mathbf{P}_{11})$, which connects endpoints of L_1 and L_2, is also in the same direction as L_2 $(\mathbf{P}_{22} - \mathbf{P}_{21})$.

$$\mathbf{P}_{22} - \mathbf{P}_{21} = k(\mathbf{P}_{21} - \mathbf{P}_{11}) \qquad \text{where k is a constant} \qquad (6.19)$$

To check link collision, we need to find if the endpoints of L_1, $\mathbf{P}_{11}$ or $\mathbf{P}_{12}$, are on L_2. Substituting these two points successively as $\mathbf{P}_2$ in (2), we can solve for s_2.

$$s_2 = (\mathbf{P}_{12} - \mathbf{P}_{21}) / (\mathbf{P}_{22} - \mathbf{P}_{21}) \qquad (6.20)$$

or $$s_2 = (\mathbf{P}_{11} - \mathbf{P}_{21}) / (\mathbf{P}_{22} - \mathbf{P}_{21}) \qquad (6.21)$$

If $0 \le s_2 \le 1$, collision happens. When (6.13) is true and (6.19) is false, L_1 and L_2 are parallel. Solving (6.7), we can express s_2 in term of s_1 as:

$$s_2 = (c_1 - a_1 s_1) / b \tag{6.22}$$

For every s_1 in (6.22), there is an s_2. Thus, there are an infinite number of common normal lines. For link collision, we need to check the common normal at the endpoints. Therefore we should use $s_1 = 0$ or $s_1 = 1$ in (6.22) to check if s_2 is within the range. The whole collision detection algorithm is summarized in Figure 6.5 for cases when the two lines containing L_1 and L_2 are collinear, parallel, intersecting, or skew.

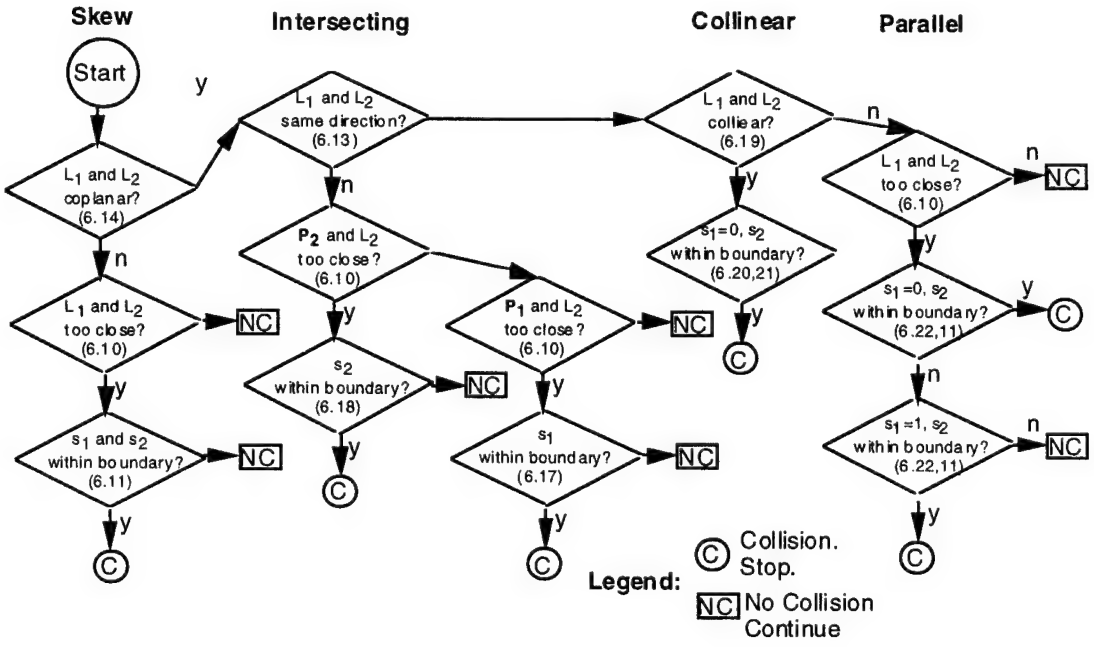

Figure 6. 5 The collision detection algorithm

6.2. Collision Detection of a Linkage Robot

The linkage robot, as shown in Figure 2.1, has 6 links and C_6^2=15 pairs of links. To investigate its collision detection, one pair of links is investigated at a time. Among the 15 pairs, six pairs consist of adjacent links, and they can be avoided by joint stops. Therefore, the possible collision of the remaining nine pairs should be investigated using the algorithm in Figure 6.5.

A MATLAB program was developed to execute the collision detection. Figure 6.6 shows a sample case of the program. The 'Simulate' button is to start the collision

detection, and the 'Close' button is to terminate the simulation. 'Start at', 'End at', 'Approach', 'View' are editable buttons. The tool trajectory can be modified by editing 'Start at' and 'End at' buttons. The tool orientation is decided by editing the 'Approach' button'. The 'View' button is to change the view direction. The 'Info' button is to explain this program.

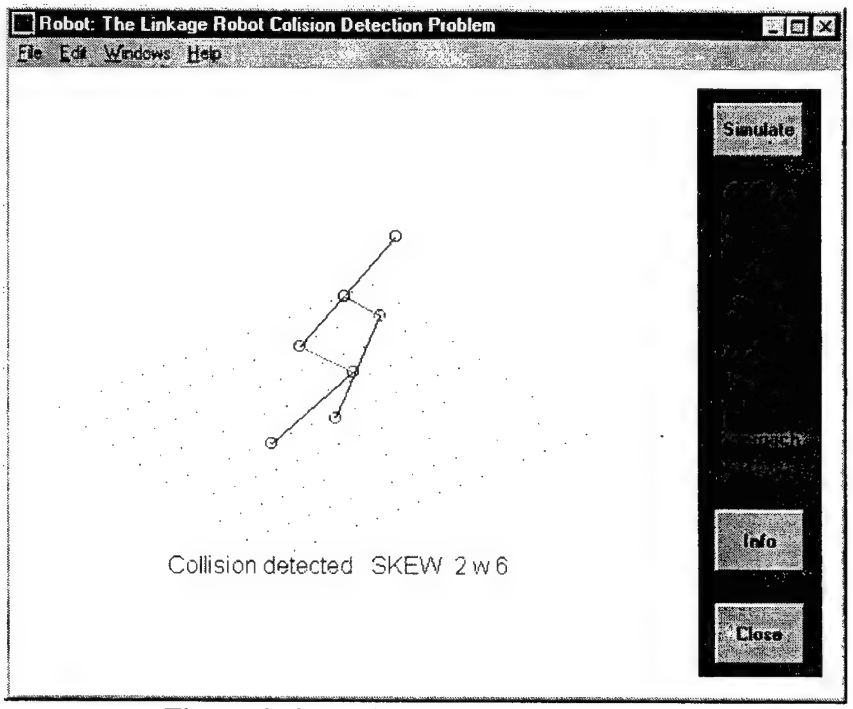

Figure 6. 6 A collision detection program

6.3 Discussion

The collision detection algorithm of serial-parallel robots can be easily modified to detect collisions of two coordinating robots. There are three moving links in each robot, and a total of six moving links need to be considered. Because the algorithm is efficient, collision detection of multiple robots can be accomplished effectively. To detect collision of three robots sharing the common workspace, the collision between two robots will be checked first, and then the algorithm will be applied to the other two pairs of robots.

CHAPTER SEVEN

SUMARY AND DISCUSSION

A linkage robot is built, analyzed, and graphically simulated in this research. The position, velocity, and force analysis of the linkage robot was derived in this project. The force and velocity analysis is based on the wrench of the screw theory, and this can be easily extended to other serial-parallel manipulators like a three-articulated-arm robot.

A prototype linkage robot was built at a cost of over $5,000. A teach pendant can control the motion of the robot, but the Rhino's controller has difficulties accepting the user's own control programs because the download software has errors.

MATLAB programs were written to verify the rate control algorithms, to simulate motion of continuous paths, and to detect possible collisions. MATLAB's strength is matrix manipulation, and it is ideal for this robotic control.

MATLAB graphic simulation is limited to a stick model of robots, and sometimes is confusing because of viewing directions. For future simulation work, Working Model 3D [Knowledge Revolution 1996] should be considered because of its ease of use and its real time link with MATLAB. Working Model 3D can also provide built-in collision detection.

REFERENCES

1. Fichter, E.F., 1986, "A Stewart Platform-Based Manipulator: General Theory and Practical Construction," *Int. J. of Robotics Research*, Vol. 5, No.2, pp. 157-182.

2. Gosselin, C and Angeles, J., 1990, "Singularity Analysis of Closed-Loop Kinematic Chains," *IEEE Tran. on Robotics and Automation*, Vol. 6, No.3, pp. 281-290.

3. Hunt, K.H., 1978, *Kinematic Geometry of Mechanism*, Oxford: Clarendon Press.

4. Hunt, K.H., 1983, "Structural Kinematics of In-Parallel-Actuated Robot-Arms," *Mechanisms, and Machine Theory*, Vol. 28, No. 1, pp. 31-42.

5. Knowledge Revolution, *Working Model User's Manual*, San Mateo, CA, 1996

6. Merlet, J-P, 1989, "Singularity Configurations of Parallel Manipulators and Grassmann Geometry," *Int. J. of Robotics Research*, Vol. 8, No.5, pp. 45-56.

7. Mohammadi, H.R., et. al., 1993, "On Singularity of Planar Parallel Manipulators," *Advances in Design Automation*, DE Vol. 65-1, pp. 593-600.

8. Nakamura, Y., 1991, *Advanced Robotics: Redundancy and Optimization*, Addison Wesley.

9. Notash, L., P. Podhorodeski R., 1994, "Complete Forward Displacement Solutions for a Class of Three-Branch Parallel Manipulators," *Journal of Robotic System,* Vol. 11, No. 6, pp. 471-485.

10. Paul, R.P., 1981, *Robot Manipulators: Mathematics, Programming, and Control*, The MIT Press.

11. Pennock, G.R, and Ryuh, B.S., "Dynamic Analysis of Two Cooperating Robots," *Proc. 1987 ASME Design Tech. Conf.* Boston, Ma, Sept. 1987, Vol. 2, pp. 63-72.

12. Stewart, D., 1965, "A Platform with Six Degrees of Freedom," *Proc Institute of Mechanical Engineering* Vol. 18, No 15, pp. 371-386.

13. Waldron, K.J., Wang, S-L., and Bolin, S.J., 1985, "A Study of the Jacobian Matrix of Serial Manipulators," *ASME J. of Mechanisms, Transmissions, and Automation in Design*, Vol. 107, No. 2, pp.230-238.

14. Waldron, K.J. and Hunt, K.H., 1988, "Series-Parallel Dualities in Actively

Coordinated Mechanisms," *Robotic Research - The 4th International Symposium*, ed. Bolles and Roth, MIT Press, pp. 175-181.

15. Wang, S-L. and Waldron K.J., 1987, "A Study of the Singular Configurations of Serial Manipulators," *ASME J. of Mechanisms, Transmissions, and Automation in Design*, Vol. 109, No.1, pp.14-20.

16. Wang, S-L., 1991, "A Spatial Linkage for a 6 DOF Semi-Direct Drive Robot," *International Journal of Robotics and Automation*, Vol. 6, No. 1, pp. 19-24.

17. Wang, S-L. and You, J., 1992, "Motion Capability of a Six-bar Linkage Robot" *Proceedings of IASTED International Conference on Control and Robotics,* in Vancouver, Canada, August 4-7, 1992, pp. 145-7.

18. Wang, S-L., 1994, "Singularity and Duality of Parallel Robots" presented at ASME Mechanisms Conference, Minneapolis, MN, September 11-14, 1994, in *Robotic: Kinematics, Dynamics and Controls,* ed. G.R. Pennock, ASME DE-Vol. 72, pp. 451-458.

19. Yang Po-hua , et. al., 1995, "Design of a Three Degree-of-Freedom Motion Platform for a Low-Cost Driving Simulator," *Proceedings of 4th National Applied Mechanisms and Robotics Conference*, Cincinnati, OH, pp. AMR 95-080-01 to 05.

```
        'String',textEStr);
 btnPos=[xPos yPos-spacing btnWid btnHt/2];
 EeditHndl=uicontrol( ...
        'Style','edit', ...
        'Units','normalized', ...
        'Position',btnPos2, ...
        'String',editEStr);

%====================================
% The View Direction button
%====================================
 btnNumber=4;
 yPos=0.90-(btnNumber-1)*(btnHt+spacing);
 textVStr='View';
 editVStr='-3 -4 5';

%      View button information
 btnPos1=[xPos yPos-spacing+btnHt/2 btnWid btnHt/2];
 btnPos2=[xPos yPos-spacing btnWid btnHt/2];
 VeditHndl=uicontrol( ...
        'Style','text', ...
        'Units','normalized', ...
        'Position',btnPos1, ...
        'String',textVStr);
 btnPos=[xPos yPos-spacing btnWid btnHt/2];
 VeditHndl=uicontrol( ...
        'Style','edit', ...
        'Units','normalized', ...
        'Position',btnPos2, ...
        'String',editVStr);

%====================================
% The Approach direction button
%====================================
 btnNumber=5;
 yPos=0.90-(btnNumber-1)*(btnHt+spacing);
 textAStr='Approach';
 editAStr='1 0 0';

%      Approach direction button information
 btnPos1=[xPos yPos-spacing+btnHt/2 btnWid btnHt/2];
 btnPos2=[xPos yPos-spacing btnWid btnHt/2];
 AeditHndl=uicontrol( ...
```

```
        'Style','text', ...
        'Units','normalized', ...
        'Position',btnPos1, ...
        'String',textAStr);
 btnPos=[xPos yPos-spacing btnWid btnHt/2];
 AeditHndl=uicontrol( ...
        'Style','edit', ...
        'Units','normalized', ...
        'Position',btnPos2, ...
        'String',editAStr);

%====================================
% The INFO button
%====================================
 labelStr='Info';
 callbackStr='lrobot("info")';
 infoHndl=uicontrol( ...
        'Style','push', ...
        'Units','normalized', ...
        'Position',[xPos 0.20 btnWid 0.10], ...
        'String',labelStr, ...
        'Callback',callbackStr);

%====================================
% The CLOSE button
%====================================
 labelStr='Close';
 callbackStr='close(gcf)';
 closeHndl=uicontrol( ...
        'Style','push', ...
        'Units','normalized', ...
        'Position',[xPos 0.05 btnWid 0.10], ...
        'String',labelStr, ...
        'Callback',callbackStr);

%====================================
% Set the figure visible after all buttons are defined.
%====================================
 hndlList=[startHndl SeditHndl EeditHndl VeditHndl AeditHndl infoHndl closeHndl];
 set(figNumber, ...
        'Visible','on', ...
        'UserData',hndlList);
 watchoff(oldFigNumber);
```

```
 figure(figNumber);

elseif strcmp(action,'simulate'),
 axHndl=gca;
 figNumber=gcf;
 hndlList=get(figNumber,'Userdata');
 startHndl=hndlList(1);
 SeditHndl=hndlList(2);
 EeditHndl=hndlList(3);
 VeditHndl=hndlList(4);
 AeditHndl=hndlList(5);
 infoHndl=hndlList(6);
 closeHndl=hndlList(7);
 set([startHndl closeHndl infoHndl SeditHndl EeditHndl VeditHndl
AeditHndl],'Enable','off');
 set(axHndl,'Userdata',play);


%      ===== Start of Simulation
%      This is the main program for the Linkage Robot Simulation Problem
%      This function makes use of the following other functions:
%      inverse9.m(A user defined program) and helpfun.m(A standard MATLAB
function)
%      =====

%      Change view direction
 VStr=get(VeditHndl,'String');
 vd=sscanf(VStr,'%f');
 view(vd);

 format long;

%      Read in data of initial position, final position, and approach vector.
 psStr=get(SeditHndl,'String');
 ps=sscanf(psStr,'%f');
 peStr=get(EeditHndl,'String');
 pe=sscanf(peStr,'%f');
 aStr=get(AeditHndl,'String');
 a=sscanf(aStr,'%f');

%      parameters of the linkage robot
 d1=1.3;      % the distance from the left wrist to hand
 d2=.7;       % the distance from the right wrist to hand
```

```
l2=1;       % lower link length
l3=1;       % upper link length
h=.6;       % the distance between the left and right shoulder
n1=10;      % the number of intervals of the straight line path

%     Mesh setting of the simulation background
X=-1:.2:2;                              %minimum: interval: maximum
Y=-.8:.2:.8;
[XX,YY]=meshgrid(X,Y);
ZZ=XX*0+YY*0;                           %mesh plane is the
zgrid=0:.2:2;
num=size(zgrid);
x1=-1;
y1=.8;
x2=2;
y2=.8;
 x1s=zgrid*0+x1;                        %mesh coordinate for vertical line
 y1s=zgrid*0+y1;
 x2s=zgrid*0+x2;
 y2s=zgrid*0+y2;
cla;
set(figNumber,...
      'Visible','on',...
      'UserData',hndlList);

plot3(XX,YY,ZZ,'c.');       %"c", cyan, "." each data point represented by a dot
hold on                     %upgrading figure without erasing
plot3(x1s,y1s,zgrid,'c.');
hold on
plot3(x2s,y2s,zgrid,'c.');
axis off
axis([-1.2 2.2 -1.0 1.0 -1 2]);
hold on
set(axHndl,...
   'Nextplot','add')
set(axHndl,'Drawmode','Fast')
drawnow
hold on

% Looping for the simulation

for k=1:n1+1
```

```
 if k>1                                                          % moving robot
  set(joHandle,'XData',d(:,1),'YData',d(:,2),'ZData',d(:,3));    %Joint position
  set(li1Handle,'XData',d(1:2,1),'YData',d(1:2,2),'ZData',d(1:2,3)) %Link segment from
  set(li2Handle,'XData',d(2:3,1),'Ydata',d(2:3,2),'ZData',d(2:3,3)) %point 1 to point 2
  set(li3Handle,'XData',d(3:4,1),'Ydata',d(3:4,2),'ZData',d(3:4,3))
  set(li4Handle,'XData',d(5:6,1),'Ydata',d(5:6,2),'ZData',d(5:6,3))
  set(li5Handle,'XData',d(6:7,1),'Ydata',d(6:7,2),'ZData',d(6:7,3))
  set(statusHandle,'String',st);        %"set", modify the figure according to new data
 drawnow
 end
%     node position
 p=ps+(k-1)*(pe-ps)/n1;
 p1=p-a*d1;
 p2=p-a*d2-[h;0;0];
 [cs,st]=inverse9(p1,p2,l2,l3);  %     function to calculate the joint angles

 if st=='out of workspace'
  break
 end

%     left arm joint position
%     p10, p11, p12, p13 are base, elbow, wrist, and tool position
 p10=[0;0;0];
 p11=[l2*cs(1)*cs(3);l2*cs(2)*cs(3);l2*cs(4)];
 p12=[cs(1)*(l2*cs(3)-l3*cs(6));cs(2)*(l2*cs(3)-l3*cs(6));l2*cs(4)+l3*cs(5)];
 p13=p;
%     check the link length and coupler length
%     ll1=sqrt((p11-p10)'*(p11-p10))     %     ll1 is the left lower arm length
%     lu1=sqrt((p12-p11)'*(p12-p11))     %     ll2 is the left upper arm length
%     lc1=sqrt((p13-p12)'*(p13-p12));          %     lc1 is the coupler arm length

%     right arm position
 p20=[h;0;0];
 p21=p20+[l2*cs(7)*cs(9);l2*cs(8)*cs(9);l2*cs(10)];
 p22=p20+[cs(7)*(l2*cs(9)-l3*cs(12));cs(8)*(l2*cs(9)-l3*cs(12));l2*cs(10)+l3*cs(11)];
 p23=p22+a*d2;
 ll2=sqrt((p21-p20)'*(p21-p20));     %
 lu2=sqrt((p22-p21)'*(p22-p21));
 la2=sqrt((p23-p22)'*(p23-p22));
 la3=sqrt((p22-p12)'*(p22-p12));

 d=[p10';p11';p12';p23';p22';p21';p20'];
```

```
  if k==1     % draw the initial position
  joHandle=plot3(d(:,1),d(:,2),d(:,3),'ro','EraseMode','background');
%       "ro" draw the line in red color and "o" sign at end points
  li1Handle=plot3(d(1:2,1),d(1:2,2),d(1:2,3),'y-','EraseMode','background');
%       "y-" draw the line in yellow color and continue line
  li2Handle=plot3(d(2:3,1),d(2:3,2),d(2:3,3),'g-','EraseMode','background'); "g", green
  li3Handle=plot3(d(3:4,1),d(3:4,2),d(3:4,3),'c-','EraseMode','background');
  li4Handle=plot3(d(5:6,1),d(5:6,2),d(5:6,3),'m-','EraseMode','background');"m",magenta
  li5Handle=plot3(d(6:7,1),d(6:7,2),d(6:7,3),'w-','EraseMode','background'); "w", white
  statusHandle=text(...
                    'HorizontalAlignment','left',...
                    'VerticalAlignment','bottom',...
                      'Position',[-0.2,0,-2.03],...
                      'EraseMode','background',...
                    'String',st);
  axis off
  drawnow
 end

%     reduce the animation speed for observation
% for jj=1:5000,
%  jjj=1+jj;
% end

 end             % end for the  "elseif strcmp(action,'simulate')"

%      activate the control button after the simulation done
 set([startHndl closeHndl infoHndl SeditHndl EeditHndl VeditHndl
AeditHndl],'Enable','on');

elseif strcmp(action,'info');
 inStr='Linkage Robot Simulation Information';
 hlpStr= ...
        ['                                                          '
         ' This program animates the straight line motion of a linkage    '
         ' robot. Clicking the animate button starts the animation with   '
         ' the default values of the program.                             '
         '                                                                '
         ' Use the "Start at", "End at", and "Approach" buttons to edit   '
        ' the start, final position and the approach vector.              '
         '                                                                '
         ' "View" button is to edit the view direction.                   '
         '                                                                '
```

```
        ' This program makes use of two other functions inverse.m:      '
        ' and helpfun.m.                                                 '];
helpfun(inStr,hlpStr);

end;            %for if strcmp(action)



function [cs,st]=inverse9(pl,pr,l2,l3);
format long

% INVERSE: This function calculates the six joint angles of the linkage robot. pl and
%       pr are the position vector of the left and right wrists. l2 and l3
%       are the link lengths of the lower and upper arms. [cs] is the output
%       vector of sin and cos functions of the six angles.

%       Angles of left arm
thl1=atan2(pl(2),pl(1));                        %theta 1
thl3s=-((pl'*pl-l2^2-l3^2)/2*l2*l3);
thl3c=sqrt(1-thl3s^2);
thl3=atan2(thl3s,thl3c);                        %theta 3

sl1=sin(thl1);
cl1=cos(thl1);
sl3=sin(thl3);
cl3=cos(thl3);
yy=pl(3)*(l2-l3*sl3)-l3*cl3*(pl(1)*cl1+pl(2)*sl1);
xx=pl(3)*l3*cl3+(l2-l3*sl3)*(pl(1)*cl1+pl(2)*sl1);
thl2=atan2(yy,xx);                              %theta 2


%       Angles of right arm
thr1=atan(pr(2)/pr(1));
thr3s=-((pr'*pr-l2^2-l3^2)/2*l2*l3);
thr3c=sqrt(1-thr3s^2);
thr3=atan2(thr3s,thr3c);
sr1=sin(thr1);
cr1=cos(thr1);
sr3=sin(thr3);
cr3=cos(thr3);
yy=pr(3)*(l2-l3*sr3)-l3*cr3*(pr(1)*cr1+pr(2)*sr1);
xx=pr(3)*l3*cr3+(l2-l3*sr3)*(pr(1)*cr1+pr(2)*sr1);
thr2=atan2(yy,xx);
```

```
sl1=sin(thl1);
cl1=cos(thl1);
sl2=sin(thl2);
cl2=cos(thl2);
sr1=sin(thr1);
cr1=cos(thr1);
sr2=sin(thr2);
cr2=cos(thr2);
sl23=sin(thl2+thl3);
cl23=cos(thl2+thl3);
sr23=sin(thr2+thr3);
cr23=cos(thr2+thr3);
cs=[cl1,sl1,cl2,sl2,cl23,sl23,cr1,sr1,cr2,sr2,cr23,sr23];

if thr3s^2>1 | thl3s^2>1,
  st='out of workspace'
else
  st=' '
end
```